Parallel Sentence Discovery for Low-Resource Languages

by

Jason R. Smith

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

May, 2013

Abstract

Statistical machine translation (SMT) accuracy depends crucially on training data in the form of parallel corpora—bilingual documents that are direct translations of each other. Most easily accessible parallel corpora originate from government and news sources, covering a limited set of languages and domains. To expand coverage, we can discover parallel sentences in comparable corpora—bilingual documents about the same topic that are not direct translations. Though comparable corpora are extremely varied, we can exploit their lexical and structural cues to extract parallel sentences that improve machine translation. We demonstrate this on three representative parallel corpora: the Web, Twitter, and Wikipedia. We show in each case how their unique structure reveals signals that can be exploited to discover parallel sentences cheaply and effectively. We estimate how much data can be found by language pair and domain, and we show that the discovered parallel data substantially improves SMT performance, especially for languages and domains with low coverage.

The Web is the by far the largest source of comparable data, containing trillions of words and constantly growing. The greatest challenge in mining the Web is accessing and searching over this massive amount of data. We use Amazon's Elastic MapReduce and Common Crawl, a publicly available web crawl, to scale up a previous mining approach (Resnik and Smith, 2003) to 32 terabytes of webpages. We mine 386 million tokens of parallel data for 18 language pairs in about 24 hours for under $500. The data we extracted improves SMT performance by 0.5 BLEU on news domain test sets for several language pairs, and by up to 5.0 on open domain test sets for Spanish-English.

For an enormous, unaligned collection of documents like the Web, simple heuristics are sufficient for extracting large amounts of parallel data. When we move to smaller, more structured datasets, we can afford to take advantage of that structure. The next source of data we investigate is Twitter, a microblogging service where users post tweets (140 character messages) to their followers. Parallel data on Twitter may arise intentionally from bilingual users tweeting the same content in two languages, or incidentally from different users referencing the same current event. We analyze several potential signals for finding parallel data: hashtags, user mentions, authors, and URLs. We use these signals to align potentially parallel tweets, and apply a supervised classifier to determine which pairs are actually parallel.

Some comparable corpora have stronger indicators of where parallel data can be found. Wikipedia, a multilingual online encyclopedia, is attractive as a comparable corpus because articles on the same topic are linked across languages. However, the articles are not parallel since they are rarely created by direct translation. We develop a novel sentence alignment model for these comparable article pairs which is able to capture document structure, allowing it to improve over the baseline which classifies sentence pairs independently. We extract parallel data for Spanish, German, and Bulgarian (paired with English), and see BLEU improvements of up to 10 points on test sets created from Wikipedia.

The data we collected from the Web, Twitter, and Wikipedia substantially improves SMT performance across several languages and domains. We see especially large improvements when testing on domains with low coverage. We not only find large amounts of parallel data across many language pairs and domains, we also release this data as a resource to the SMT community. Our Web data has already been used as part of the training sets for the Workshop on Statistical Machine Translation.

Chapter 1

Introduction

Statistical machine translation (SMT) systems are trained using a large collection of translated sentence pairs known as a parallel corpus. Common sources of parallel data include parliament proceedings, books, and news articles. For some language pairs, we have large amounts of this data. For example, the Canadian Hansards are parliamentary proceedings that give us millions of words of French/English parallel data (Germann, 2001a). Similarly, the proceedings of European Union parliament are a source of parallel data for all of the languages of its member states (Koehn, 2005). Outside of these government sources, we also have large collections of parallel data from news agencies for some language pairs, such as Chinese/English (Ma, 2005). However, for most language pairs, we have little to no data available. In addition, even when parallel data is available, it often does not match the domain of the data you wish to translate, which hurts translation quality (Munteanu and Marcu, 2005). Since the source of parallel corpora are institutions that want to disseminate information in multiple languages, there will always be a bias in the genre of the available data. Governments, news organizations, and international companies are the most common sources of parallel data, giving us large amounts of text in more formal domains, but leaving us with little in informal domains such as the short, conversational text found on Twitter. Figure 1.1 gives some example parallel sentences from Europarl as well as Twitter, showing the drastic difference in vocabulary and syntax that can be found across domains.

Europarl	
English Spanish	Please rise, then, for this minute' s silence. Invito a todos a que nos pongamos de pie para guardar un minuto de silencio.
English Spanish	Madam President, on a point of order. Seora Presidenta, una cuestin de procedimiento.
Twitter	
English Spanish	can u listen to my song please? puede escuchar mi canción?
English Spanish	RT @celtics : Kobe Bryant surpassed Michael Jordan as the all-time NBA #AllStar scorer tonight. #KobeBryant más puntos en #AllStar games de todos los tiempos. :')

Figure 1.1: Parallel sentences from the European parliament and Twitter. The Twitter data was selected from automatically mined data (see Chapter 5). While the Europarl sentences are well-formed and grammatical, the sentences found on Twitter contain abbreviations, lack punctuation, and contain markup and emoticons. In addition, the sentence pairs we find on Twitter are not always exact translations, though we still may find them to be useful as parallel data. We will explore this topic further in Section 1.1.

[A point about motivation that doesn't quite come out here. Broadly, there are two main customers for translation: those interested in disseminating information in multiple languages (governments, news organizations, and international companies), and those interested in assimilating information from multiple languages (intelligence agencies, travelers, consumers, researchers). The vast bulk of data that we get is from the former, who generate it naturally in the course of their day-to-day activities – but their interests do not always coincide with the needs of the latter, hence there is a mismatch between the data we have, and the data that we want. The easy to get data is what you get from web scraping. However, to get data for the other customer base, you need to look for other signals —evidence that people are talking about the same thing in multiple languages.] **[Added something in the first paragraph about this. –JS]**

The creation of new parallel corpora can be expensive, especially when bilingual speakers are rare for the language pair of interest. Germann (2001b) investigated the costs of collecting enough data to build Tamil/English SMT system. They found that professionally translated data would cost $0.36 per word. Germann (2001b) and others (Zaidan and Callison-Burch, 2011) were able to reduce the cost of creating parallel corpora by looking to non-professional translators, but the cost is still around $0.10 per word. In order to acquire more parallel data without this cost, researchers have looked to multilingual corpora which share some content across languages, but are not directly translated. Such corpora are referred to as comparable corpora, and examples include multilingual news feeds (Munteanu and Marcu, 2005), Wikipedia articles (Adafre and de Rijke, 2006; Smith et al., 2010) (see Figure 1.2), and the Web (Resnik, 1999; Nie et al., 1999; Chen and Nie, 2000).

Antipartícula

A cada una de las partículas de la naturaleza le corresponde una **antipartícula** que posee la misma masa, el mismo espín, pero distinta carga eléctrica. Algunas partículas son idénticas a su antipartícula, como por ejemplo el fotón, que no tiene carga. Pero no todas las partículas de carga neutra son idénticas a su antipartícula.

Antiparticle

From Wikipedia, the free encyclopedia

Corresponding to most kinds of particles, there is an associated **antiparticle** with the same mass and opposite electric charge. For example, the antiparticle of the electron is the positively charged antielectron, or positron, which is produced naturally in certain types of radioactive decay.

Figure 1.2: An example of a Spanish/English document pair from Wikipedia.

The distinction between parallel and comparable corpora is not always clear. Even corpora which are generally considered as parallel require some amount of processing to find parallel sentences. A translator may chose to translate a compound sentence as two sentences, or vice-versa, so naively assuming that sentences are aligned in order will not work. Also, there may be large insertions or deletions of sentences even in curated sources of parallel data, such as the Canadian Hansards (Gale and Church, 1993; Chen, 1993). Sentence-aligning these corpora does not require existing parallel data or a bilingual dictionary for the language pair of interest. Instead, the structure of the documents and the lengths of the sentences are used to determine the sentence alignment. When considering multilingual corpora which are less closely related, additional signals must be used to find parallel sentences. For corpora which are topic-aligned but not directly translated, lexical information must be used to determine which sentence pairs should be aligned (Munteanu and Marcu, 2005). When comparable corpora are not topic-aligned, other signals such as URLs and hyperlinks are exploited to find plausible document alignments (Resnik and Smith, 2003). [This paragraph feels out of place. Try reading the chapter, skipping this one completely. There's a pretty nice flow from the previous to the next.] **[I moved it up and modified it to fit better. –JS]**

To give a better idea of the variety of comparable corpora, we will consider the categorization of multilingual corpora given by Fung and Cheung (2004a):

1. Parallel corpus: A sentence-aligned corpus containing bilingual translations of the same document. Curated parallel corpora fall in this category.

2. Noisy parallel corpus: A corpus containing non-aligned sentences that are nevertheless mostly bilingual translations of the same document. The Canadian Hansards, Europarl, and most other parallel corpora are not strictly parallel until some processing is done.

3. Comparable corpus: A corpus of non-aligned and non-translated documents which are topic-aligned. Examples include multilingual social media discussions linked by topic, and Wikipedia.

4. Quasi-comparable corpus: A multilingual corpus which is not sentence-aligned, translated, or topic-aligned. This includes multilingual news feeds and the Web.

This is just one way of categorizing comparable corpora—there are concievably many others. What is relevant for us is how the structure of different comparable corpora allow us to find parallel data. For example, multilingual news feeds have time stamps which help us align potentially parallel documents, while Wikipedia contains links between articles on the same topic across languages. The methods we use for mining parallel data will depend primarily on the signals available.

We will examine a representative set of comparable corpora: the Web, Twitter, and Wikipedia; describe the different signals used to identify parallel data, and demonstrate how extracted parallel data from these corpora improve SMT performance across several language pairs and domains. First, we scale up previous Web mining methods (Resnik and Smith, 2003) to several terabytes of data (Chapter 3). We also present a novel mining approach for Twitter, making use of metadata unique to the microblogging medium (Chapter 5). Finally, we introduce a new sentence alignment model for mining parallel data from Wikipedia which takes advantage of its fine-grained topic alignment (Chapter 4). **[I will come back to this ordering as I get to the other chapters. –JS]**

1.1 What counts as parallel?

We are concerned with finding parallel data—bilingual sentence pairs which convey the same meaning. Unfortunately, it is extremely difficult, if not impossible, to determine whether or not two sentences in different languages have the same meaning. One language may contain gender markings that the other does not, or the connotation of a word may be difficult to express in another language. Some examples of the ambiguity inherent in translation are explored by Kay (1997), one of which is shown in Figure 1.3. Even ignoring the cross-lingual issues, comparing the meaning of two sentences in the same language is still quite difficult—this is essentially the task of recognizing textual entailment (RTE) (Dagan et al., 2010).[1] SMT evaluation metrics (Papineni et al., 2002; Banerjee and Lavie, 2005; Snover et al., 2006) must also address this problem when comparing a hypothesis translation to a reference.

French	Ils signeront le document pourvu que leur gouvernement accepte.
English	They will sign the document supplied that their government accepts.
English	They will sign the document provided that their government accepts.
English	They will sign the document on condition that their government accepts.

Figure 1.3: An example of a French sentence with multiple potential translations into English taken from Kay (1997).

When evaluating methods for finding parallel data, we can either measure intrinsic or extrinsic performance. Intrinsic evaluation directly measures the quantity and quality of parallel data we extract, while extrinsic evaluation is only concerned with how the new parallel data improves SMT performance. In order to perform intrinsic evaluation, we need some criteria for determining whether or not a bilingual sentence pair is parallel. This is easy if we use parallel data, but it is preferable to evaluate our methods on the same corpora that we are extracting data from. When designing the criteria for judging parallel sentences, we focus on our extrinsic goal: improving SMT performance. If a sentence pair is likely to improve performance when added

[1] RTE involves determining whether or not one piece of text can be inferred from another.

to our SMT system's training data, we would like to extract it. The details of our annotation criteria can be found in Chapter 4, but in all cases they are motivated by SMT performance. To understand what will influence performance, we need to understand modern SMT systems.

1.2 Statistical Machine Translation

While machine translation has been around in some form for many decades (Locke and Booth, 1955), statistical machine translation began with the work of Brown et al. (1988, 1990, 1993). While SMT systems have evolved greatly since then, they all share some key characteristics in how they use parallel data: [This feels a little opaque at this point: either compress to lose unnecessary details, explain the difference in word-based, phrase-based, and syntax-based, give some examples, or move this to a later point.] **[I compressed it. It doesn't seem necessary to go into details unless I want to make a very specific point about how a system might handle a messy parallel sentence pair. –JS]**

1. A large collection of parallel sentences are used as training data.

2. For each parallel sentence, word-to-word correspondences are found. This step is called word-alignment, and it is usually done with unsupervised methods (Brown et al., 1993; Vogel et al., 1996).

3. Pairs of phrases, or other multi-word units, are extracted from the word-aligned

sentence pairs to form a translation model. Figure 1.4 gives an example of an extracted phrase pair.

4. A language model is created from large amounts of monolingual data in the "target" language (the language which text is translated into). This includes the target side of the parallel training data.

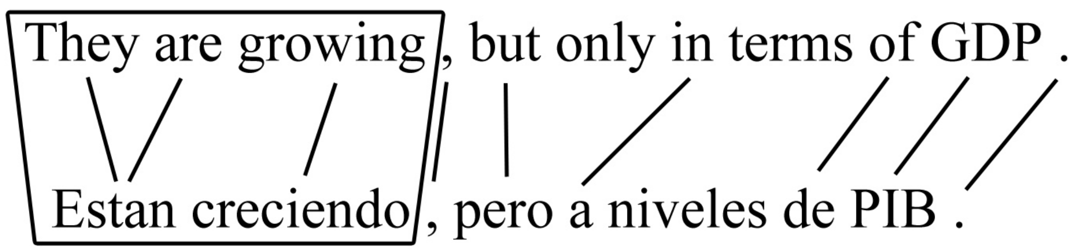

Figure 1.4: A word-aligned sentence with an extracted phrase pair.

There are additional details in each model, but the main effects of adding new parallel data are additional inputs to the translation and language models.

1.3 Evaluation Pipeline

Our evaluation setup is identical across chapters—we start with initial data that includes some standard parallel and monolingual corpora commonly used for translation. We also have extracted parallel data that we find in a comparable corpus. Table 1.3 describes how we use this data to measure SMT improvements:

	Parallel	Monolingual
Baseline	Initial	Initial
Baseline + Monolingual	Initial	Initial + Extracted
Experimental	Initial + Extracted	Initial + Extracted

Table 1.1: Parallel and monolingual data used in our SMT experiments.

In both the "Baseline + Monolingual" and "Experimental" conditions, we include the target side of the extracted parallel sentences in the monolingual training data. We do this to ensure that any increase in performance is coming from the parallel data. It would be simple to add monolingual text from a comparable corpus to an SMT system.

In all experiments, the BLEU metric (Papineni et al., 2002) is used to evaluate SMT performance. It is the most commonly used evaluation metric, allowing us to easily compare with other work. The BLEU metric combines n-gram precision (the percentage of n-grams in the hypothesis translation which are found in the reference) with a brevity penalty. The exact formula is:

$$\exp(\min(1 - \frac{r}{c}, 0) + \frac{1}{4} \sum_{n=1}^{4} \log p_n) \tag{1.1}$$

where r and c are the reference and hypothesis lengths, and p_n is the n-gram precision. The initial data, test sets, and other details vary by experiment.

Chapter 2

Previous Work

There have been several techniques developed for finding parallel data in a wide range of multilingual corpora. Here we will review previous work in this area in order to compare with the work we will present in later chapters. We will start with the earliest work—the alignment of manually translated corpora. [Rather than think of this as a compendium of related work, it might be useful to think of a more goal-oriented chapter: setting the stage for the rest of your dissertation. What did we know or not know at the outset?]

2.1 Parallel Corpus Alignment

Research in SMT began as large parallel corpora became available (Brown et al., 1988, 1990, 1993). These corpora include the Canadian Hansards (French-English parliament proceedings) and the Hong Kong Laws Corpus, among many others. While these corpora were parallel in the sense that they were created by directly translating text in one language, they were not sentence aligned. Noise in the form of missing data or sentences without a 1 : 1 correspondence made alignment a non-trivial problem. This lead to the development of several approaches for aligning parallel corpora in the early 1990s. Since the problem of aligning noisy-parallel corpora is closely related to finding parallel sentences in comparable corpora, we will give an overview of these approaches.

Perhaps the most well known work on parallel corpus alignment is Gale and Church (1991, 1993). The authors described a sentence alignment method based on dynamic programming which used only sentence length to determine whether or not two sentences were parallel. This method is widely applicable since it assumes almost no linguistic knowledge.[1] Despite this, it achieved very high accuracy on a corpus of economic reports from the Union Bank of Switzerland in English, French and German. Brown et al. (1991) had a similar approach, using only sentence lengths to align parallel corpora, but they measured length in words rather than characters.

Even when there is no bilingual lexicon available for a language pair, if the source and target languages are similar enough it may be possible to use the surface similarity of words to infer cognates. Simard et al. (1993) made use of this by replacing the length based alignment scoring of Gale and Church (1993) with a cognate based

[1] The only bit of information about the language pair required is a ratio of sentence lengths in characters.

scoring method using a simple method for identifying cognates. Church (1993) made use of cognates with a radically different approach: creating a dotplot of character n-gram matches weighted by inverse frequency, and then finding an alignment which best matches the dots. While this cognate based approach was intended to work for similar languages, the authors noted that even in language pairs like Japanese-English, matches can be found on technical terms and markup.

The sentence alignment approach of Kay and Röscheisen (1993) also used little linguistic knowledge, though they build a bilingual dictionary from the parallel text to facilitate alignment. Beginning with an initial set of sentence alignments, they iteratively update the bilingual dictionary and the sentence alignments in a manner similar to Viterbi EM, though no explicit probability model is given. Chen (1993) had a similar approach, except he incorporated the learning of both sentence and word alignments into a probabilistic model. While this is similar to our work in that there is a generative story of document pairs used to infer sentence alignments, Chen (1993) used a joint probability distribution of source/target sentence pairs which must be approximated for efficient inference, and several choices are made in the inference strategy which assume a strongly monotonic sentence alignment. Stochastic Viterbi EM is used to find the best sentence alignment.

As an alternative method for creating a bilingual dictionary, Fung and Church (1994) built a vector for each source/target word representing how it is distributed in the parallel corpus. The intuition was that since the alignment between the source and target data was strongly monotonic, so words that appear in the same relative positions in the source/target corpora are likely to be translations of one another.

Moore (2002) builds off of the length based alignment approach of Gale and Church (1993) by adding a bootstrapping step after the initial alignment. First, a length based sentence alignment is done on the parallel corpus. Then, the sentences found to be parallel are used to train a word alignment model (IBM Model 1), and the sentence alignment dynamic program is repeated using the word alignment scores in addition to length based scores. This bootstrapping approach is popular in work on mining noisy parallel/comparable corpora (see Section 2.2).

2.2 Comparable Corpus Mining

In addition to aligning parallel texts, there has also been a considerable amount of work done on finding parallel sentence pairs in comparable corpora. A comparable corpus is a multilingual collection of documents which may contain parallel sentences, but is not completely parallel. This broad definition includes both weakly aligned data such as timestamped multilingual news feeds, and Wikipedia articles linked at the document level. Depending on the type of comparable corpus, different methods may be more or less effective for finding parallel sentences. We will split our review of comparable corpora mining methods into two categories. In Section 2.2.1, we will examine methods used on closely aligned comparable corpora, and in Section 2.2.2

we will review work on extracting parallel sentences from less related multilingual documents.

2.2.1 Noisy Parallel Corpora

The first category of work on comparable corpora mining that we will review is on noisy parallel data. While even corpora called "parallel" contain some noise, we are refering to corpora which the methods in Section 2.1 would fail on.

Similar to the dynamic programming approaches explored in Section 2.1, Zhao and Vogel (2002) used a dynamic programming strategy for aligning parallel sentences in a document pair. They create a probabilistic model of a comparable document pair $P(S,T,A)$ and choose an alignment to maximize the probability of the observed source and target documents. To estimate the probability of two sentences being aligned, they used and IBM-style word alignment models (Model 3, specifically) which were estimated on existing parallel data. Zhao and Vogel (2002) also describes a bootstraping approach where high confidence sentence alignments are added to the training data for the word alignment model, and then sentence alignments are recomputed. Much of the work on noisy parallel/comparable corpora mining used this technique (Fung and Cheung, 2004a,b; Wu and Fung, 2005; Munteanu and Marcu, 2005).

2.2.2 Comparable Corpora

In comparable corpora such as bilingual news feeds or websites, the document alignment is often not given.[2] First, we will review methods for finding comparable document pairs in a comparable corpus, and then methods for identifying parallel sentence pairs within these documents.

2.2.2.1 Finding Comparable Document Pairs

The Gigaword corpus contains news feeds in multiple languages, and is annotated with the date of publication. Since these news articles are potentially on the same topic, there are potentially parallel sentence pairs in these articles. Munteanu et al. (2004); Munteanu and Marcu (2005); Fung and Cheung (2004a,b) make use of this information to find comparable document pairs. The basic strategy is to first consider all bilingual article pairs published within a time window to be potentially comparable. Then, documents in one language are projected through a bilingual dictionary, and bag-of-words based document similarity measures are used to prune this large set of document pairs. This requires either existing parallel data or at least a bilingual dictionary. Document pairs that pass through these filter are then mined for parallel sentences.

Multilingual websites are another potential source for comparable or parallel document pairs. STRAND (Resnik and Smith, 2003) used some heuristics for identifying

[2] A notable exception to this is Wikipedia

links between versions of the same website in different languages. This provides a candidate set of document pairs, which are further filtered by looking at their HTML structure. Each website is converted into a list of start tags, end tags, and "chunks" (text within a tag), and these lists are aligned using standard dynamic programming techniques. This alignment is not only used to determine whether a pair of websites is comparable, but it also gives an alignment of text chunks which greatly narrows down the space of possible sentence alignments

A drastically different approach for finding parallel web pages is given by Uszkoreit et al. (2010). Using a existing language identification and translation systems, they identify the language of all webpages and translate the non-English ones into English. Since all documents are now in the same language, the problem of identifying comparable webpages is treated as near-duplicate detection. An index is built mapping n-grams to documents, and this index is used to find a bag-of-n-grams score for potentially comparable documents. The computation is kept feasible by only creating index entries for rare n-grams.

Ture and Lin (2012) used cross-lingual information retrieval techniques to find comparable document pairs in Wikipedia. While Wikipedia already provides annotated comparable document pairs through interwiki links, the authors consider all possible German-English article pairs as potentially containing comparable data.

2.2.2.2 Finding Parallel Sentences

Once comparable document pairs have been identified, most comparable corpora extraction methods will independently judge each sentence pair as parallel or non-parallel. Since there is often a very large amount of document pairs and thus potential sentence pairs, filters are used to prune out sentence pairs that are highly unlikely to be parallel. For example, Munteanu and Marcu (2005) used a sentence length filter to remove sentence pairs where one sentence was more than twice as long as the other. In addition, they used a word overlap filter based on the bilingual dictionary used to find candidate document pairs.

Given a filtered set of sentence pairs, more expensive methods of scoring sentence pairs can be used. Munteanu and Marcu (2005) use a binary MaxEnt classifier to ultimately determine whether or not a sentence pair is parallel. The classifier is trained on parallel data and makes used of features which are mostly based on word alignments. Others Fung and Cheung (2004a,b); Tillmann (2009); Tillmann and Xu (2009) use a single score for sentence pairs based on either a word alignment model or bag-of-words similarity after projection through a bilingual lexicon, and tune a threshold on held out data.

Chapter 3

Examining the Usefulness of Comparable Corpora

[1]

Comparable corpora are multilingual collections of documents which are not strictly parallel, but may contain some parallel data. This includes everything from multilingual news feeds, which may not even be topic-aligned, to Wikipedia, which has fine-grained topic alignment across languages. This chapter will examine the feasibility of extracting large amounts of parallel data from comparable corpora, with and without existing resources, and analyze the effect this data has on end-to-end SMT performance. Two comparable corpora are used here: CommonCrawl, a publicly available crawl of the entire Web, and Wikipedia, an online collaborative encyclope-

[1]This chapter continues the work done with my team at the 2012 Machine Translation Marathon. I would like to thank Philipp Koehn for performing SMT experiments, and my teammates Herve Saint-Amand and Magdalena Plamada for both coding and analysis of the parallel data.

dia. We conduct an extensive empirical exploration of the mined data, demonstrating coverage in a wide variety of languages and domains (§??). Even without extensive pre-processing, the data improves translation performance on strong baseline news translation systems in five different language pairs (§??). On general domain and speech translation tasks where test conditions substantially differ from standard government and news training text, web-mined training data improves performance substantially, resulting in improvements of up to 1.5 BLEU on standard test sets, and 5 BLEU on test sets outside of the news domain.

3.1 CommonCrawl

A promising source of parallel data is the Web, as many websites are presented in multiple languages. Researchers have been exploring ways to mine the Web for parallel data for over a decade (Resnik, 1999; Nie et al., 1999; Chen and Nie, 2000). One major challenge is access to the data - large companies such as Google regularly maintain a crawl of the entire Web, but even storing that much data may not be possible on a university's local cluster. In this work we are interested in methods for mining parallel data which are feasible for researchers in academia to use. For this reason we look at the CommonCrawl corpus, which is a publicly available crawl of the web created by the CommonCrawl foundation.[2] As a baseline approach to both document alignment and sentence alignment, we apply the STRAND algorithm

[2] `commoncrawl.org`

(Resnik and Smith, 2003) to this dataset.

The CommonCrawl corpus is hosted on Amazon's Simple Storage Service (S3). It can be downloaded to a local cluster, but the transfer cost (roughly 10 cents per gigabyte, leading to a total of $81,000[3]) is prohibitive. However, the data can be accessed freely from Amazon's Elastic Compute Cloud (EC2) or Elastic MapReduce (EMR) services, which charge modest fees for compute time, but allow free access to data which is alread on S3. In our pipeline, we perform the first step of identifying candidate document pairs using Amazon EMR, download the resulting document pairs, and perform the remaining steps on our local cluster. We chose EMR because our candidate matching strategy fit naturally into the Map-Reduce framework (Dean and Ghemawat, 2004).

3.1.1 STRAND Pipeline

The following is the pipeline we use for our STRAND (Resnik and Smith, 2003) baseline: **[TODO: STRAND diagram –JS]**

1. Candidate pair selection: Retrieve candidate document pairs from the CommonCrawl corpus

2. Structural Filtering:

   (a) Convert the HTML of each document into a sequence of start tags, end

[3] `http://aws.amazon.com/s3/pricing/`

tags, and text chunks

   (b) Align the linearized HTML of candidate document pairs

   (c) Decide whether to accept or reject each pair based on features of the alignment

3. Segmentation: For each text chunk, perform sentence and word segmentation

4. Sentence Alignment: For each aligned pair of text chunks, perform the sentence alignment method of Gale and Church (1993)

5. Sentence Filtering: Remove sentences which appear to be boilerplate

3.1.1.0.1 Candidate Pair Selection

Here we describe the Map-Reduce job which identifies candidate parallel websites. We adopt a strategy similar to that of Resnik and Smith (2003) for finding candidates in the Internet Archive.

Map-Reduce (Dean and Ghemawat, 2004) is a framework for massively distributed parallel programming over large amounts of data. It includes two main steps: a Map step which creates key-value pairs from the input entries, and a Reduce step which produces output from each key paired with all matching values.

The mapper operates on each website entry in the CommonCrawl data. It scans the URL for some indicator of its language. Specifically, we check for:

1. Two/three letter language codes (ISO-639)

2. Language names in English and the language of origin

If any of these codes or names are present in a URL and surrounded by non-alphanumeric characters (for example: `www.website.com/fr/`), this will be seen as a potential match. The mapper will then output the following (key, value) pair:

- Key: `www.website.com/*/`
- Value: `www.website.com/fr/`, French, (full website entry)

[TODO: detailed matching information, with code –JS]

The reducer will then receive all websites mapped to the same "language independent" URL. If two or more websites match, the reducer will output all matching document pairs, as long as they are not in the same language, as determined by the language identifier from the URL.

This URL-based matching is a simple and inexpensive solution to the problem of finding candidate document pairs. The mapper will discard most, and neither the mapper nor the reducer do anything with the HTML of the documents aside from reading and writing them. More sophisticated approaches have been used (Uszkoreit et al., 2010; Ture and Lin, 2012), but they may be prohibitively expensive to run on Amazon, and the focus of this work is to show that mining parallel data from the entire Web can be affordable. **[1250 hours for bitvector generation –JS]**

3.1.1.0.2 Structural Filtering

A major component of the STRAND system is the alignment of HTML documents. This alignment is used to determine which document pairs are actually parallel, and if they are, to align pairs of text blocks within the documents.

The first step of structural filtering is to linearize the HTML. This means converting the DOM tree into a sequence of start tags, end tags, and chunks of text. Some tags (those that may be often found within text, such as “font” and “a”) are ignored during this step. Next, the tag/chunk sequences are aligned using dynamic programming. The objective of the alignment is to maximize the number of matching items.

Given this alignment, Resnik and Smith (2003) define a small set of features which indicate the alignment quality. They annotated a set of document pairs as parallel or non-parallel, and trained a classifier on this data. We also annotated 101 Spanish-English document pairs in this way and trained a maximum entropy classifier. However, even when using the best performing subset of features, the classifier only performed as well as a naive classifier which labeled every document pair as parallel, in both accuracy and F1. For this reason, we excluded the classifier from our pipeline.

3.1.1.0.3 Segmentation

The text chunks from the previous step may contain several sentences, so before the sentence alignment step we must perform sentence segmentation. We use the

Punkt sentence breaker from NLTK (Loper and Bird, 2002) to perform both sentence and word segmentation on each text chunk.

3.1.1.0.4 Sentence Alignment

For each aligned text chunk pair, we perform sentence alignment using the algorithm of Gale and Church (1993).

3.1.1.0.5 Sentence Filtering

Since we do not perform any boilerplate removal in earlier steps, there are many sentence pairs produced by the pipeline which contain menu items or other bits of text which are not useful to an SMT system. To remove this data, we prune segment pairs unless both segments contain at least 5 tokens composed of alphanumeric characters only, and end with punctuation. We also remove any sentence pairs which are identical.

3.1.2 Results

3.1.2.1 Intrinsic Evaluation

To evaluate the quality of the parallel data produced , we manually check a set of randomly selected 200 sentence pairs for three language pairs. The texts are very heterogeneous, covering several topical domains, such as: tourism, advertising, technical specifications, finances, e-commerce or medicine. For German-English, 78% of

Language	Precision
Spanish	82%
French	81%
German	78%

Table 3.1: Precision on the extracted parallel data for Spanish, French, and German (paired with English).

the extracted data represent perfect translations, 4% are paraphrases of each other (convey a similar meaning, but cannot be used for SMT training) and 18% represent misalignments. Furthermore, 22% of the true positives are potentially machine translations, whereas in 13% of the cases one of the sentences contains an extra tail. As for the false positives, 13.5% of them have been caused by language identification errors, the remaining ones representing failures in the alignment process. Similar tendencies have been observed in the other data sets. All in all, the precision of the mining process is on average 80% for the considered language pairs, as Table 3.1 shows. This analysis suggests that language identification and SMT output detection (Venugopal et al., 2011) may be useful additions to the pipeline.

3.1.2.2 Extrinsic Evaluation

We applied this baseline to the full 2009-2010 crawl in seven separate chunks.[4] In Table 3.2 we show SMT experiments before and after adding this data to a baseline. In both the baseline and experiments with added data we include the target side of

[4]This was to avoid out-of-memory errors which occurred when running on the full crawl in a single experiment.

the mined parallel data in the language model, to show the improvements are not just coming from the additional monolingual data. These results show substantial gains on top of an already strong SMT system.

	FR-EN	**EN-FR**	**ES-EN**	**EN-ES**	**EN-DE**
Baseline	29.88	28.50	32.80	32.83	16.61
+Web Data	30.08	28.76	33.39	33.41	17.30

Table 3.2: BLEU scores for several language pairs before and after adding the mined parallel data to a baseline system trained on data from WMT 12.

3.2 Wikipedia

Wikipedia[5] is an online collaborative encyclopedia available in over 200 languages. It is part of the larger WikiMedia project, which includes other multilingual websites such as Wiktionary and Wikinews. It is a promising source of parallel data due to the "interwiki" link structure. Each article has links to articles on the same topic in other languages. These are occasionally direct translations, but for the most part they are simply topic-aligned.

While this is of course included in the Web, the web mining techniques used in general Web mining are not appropriate for this corpus. First, URL matching techniques will not work for the majority of bilingual article pairs. Also, the article pairs in Wikipedia are not often direct translations, so methods that rely on HTML alignment, such as STRAND, are not appropriate. Since Wikipedia provides document

[5] `www.wikipedia.org`

alignment across languages via "interwiki" links, the document alignment step is unnecessary, though see Ture and Lin (2012) - the authors ignore the interwiki link structure and align documents themselves. While the links could be missing some article pairs, they are fairly well maintained by Wikipedia contributors. **[I should compare with his data if possible. –JS]**

3.2.1 Software

Here we will describe the pipeline we use for mining parallel data from Wikipedia, and show how each of the steps can be done using the WikiDumpTools[6] software package. We work with static dumps of Wikipedia to avoid constant bandwidth usage and to ensure consistent results across several experiments.

1. For each language we are working with, download the dump of all articles, the interwiki link table, and the redirect table
2. Create indexed versions of each dump for quick random access
3. Create a list of article pairs using the interwiki link tables and redirect tables for each language pair
4. Iterate through the list of article pairs, retrieve their Wikitext from the indexed dump, and output plain text document pairs

[6] TODO: Github URL

3.2.1.0.1 Downloading static dumps

Database dumps of Wikipedia can be found at `dumps.wikimedia.org`. They are listed by language code, so "enwiki" is the English Wikipedia, "eswiki" is the Spanish, etc. There are many different types of database dumps here, but the ones we are interested in are the main articles, the interwiki link table, and the redirect table. These files end in "`pages-articles.xml.bz2`", "`langlinks.sql.gz`", and "`redirect.sql.gz`", respectively.

3.2.1.0.2 Indexing static dumps

When iterating through the interwiki links, we need to quickly find articles in the database dumps by title. In order to do this, we build an indexed dump, where each entry in the index contains byte offsets for the Wikitext of an article.

```
python wdtools.py --index-wiki (dump file) --output indexed-wiki
```

The dump file should be the file that ended in "`pages-articles.xml.bz2`" This command will create two files: `indexed-wiki.index.gz`, which is the index to the dump, and `indexed-wiki.dump`, which is the uncompressed Wikitext of all articles in the original dump. This file is uncompressed to allow efficient random access.

3.2.1.0.3 Finding article pairs

The interwiki links table contains the outgoing interwiki links for the articles in each language. The redirect table contains a record of all redirect pages, which are usually spelling variants. WikiDumpTools uses the interwiki link tables and redirect tables from two languages to create a complete list of article pairs matched across languages.[7] Note that this list of article pairs is, for the most part, parallel data itself, and we explore its use in later chapters.

```
python wdtools.py --get-pairs (output file) --source (dump index),(interwiki
table),(redirect table) --target (dump index),(interwiki table),(redirect table)
```

"Source" and "target" here refer to the source and target languages. The dump indexed is required because it contains a mapping from article IDs to article titles, and the other tables make use of both.

3.2.1.0.4 Creating article pairs

Using the list of article pairs, we can now output plain text document pairs.

```
python wdtools.py --output-pairs docs-out --source-dump (indexed dump)
--target-dump (indexed dump) --pair-list (list of article pairs)
```

This will iterate through the list of article pairs, retrieve them from the indexed dumps, remove Wikitext markup, and create lists of documents in the source and target languages. Two files will be created: `docs-out.source.gz` and `docs-out.target.gz`. These files are aligned document pairs, where document boundaries are separated by empty lines. Note that no sentence breaking or word tokenization is performed.

After the steps in this pipeline, comparable corpus mining techniques can be used on the document pairs. Software for performing this step will be described in later chapters.

[7] The redirect tables are needed since the interwiki link can point to a redirect page.

3.3 Comparison to Current Parallel Resources

For several language pairs, there are parallel corpora already available. Some notable examples include Europarl (Koehn, 2005), the U.N. Corpus (Eisele and Chen, 2010). In addition to these multilingual corpora, there are many parallel corpora for specific language pairs. For the most part, the language pairs that received the most attention are European languages, Arabic, and Chinese paired with English.

In this section, we will compare the amount of existing parallel data for many language pairs with the amount of comparable data available from our two sources, CommonCrawl and Wikipedia. This is meant to give an estimate of how much parallel data we can expect to mine from these sources. To estimate the amount of existing parallel data, we use OPUS, a freely available collection of parallel data (Tiedemann, 2009). Table 3.3 shows the amount of existing parallel data for selected languages paired with English. **[I will have to double check these carefully, since there are a few corpora I know of that are missing, and older versions of Europarl are used in the counts. The counts are also dominated by the OpenSubtitles corpus, which people rarely use. –JS]**

Table 3.4 shows the amount of parallel data mined from CommonCrawl on the same language pairs using the baseline system STRAND. Table 3.5 shows an upper

	BG	**CS**	**DE**	**ES**	**FR**
Segments	18.5M	22.8M	8.3M	51.5M	39.7M
Source Tokens	137.6M	153.7M	92.1M	361.4M	247.6M
Target Tokens	163.7M	187.9M	100.9M	331.0M	235.7M
	JA	**KO**	**RU**	**UR**	
Segments	0.5M	0.1M	18.3M	1.0K	
Source Tokens	5.4M	3.8M	48.5M	2.6M	
Target Tokens	0.5M	0.4M	39.2M	10.0K	

Table 3.3: The amount of parallel data available from OPUS (Tiedemann, 2009) for each language paired with English. Source tokens are counts of the foreign language tokens, and target tokens are counts of the English language tokens.

	BG	**CS**	**DE**	**ES**	**FR**
Segments	962K	886K	8.04M	6.11M	10.9M
Source Tokens	8.72M	7.50M	83.9M	75.4M	135M
Target Tokens	8.53M	7.95M	88.4M	68.8M	121M
	JA	**KO**	**RU**	**UR**	
Segments	1.80M	787K	3.86M	59.7K	
Source Tokens	9.59M	6.57M	36.6M	828K	
Target Tokens	19.1M	7.42M	37.2M	723K	

Table 3.4: The amount of parallel data mined from CommonCrawl for each language paired with English. Source tokens are counts of the foreign language tokens, and target tokens are counts of the English language tokens.

bound on the amount of data that could be extracted from Wikipedia. The numbers shown are the sum of the number of tokens of the smaller article in each bilingual article pair.

From the annotated Wikipedia article pairs collected by Smith et al. (2010), we can estimate the relationship between the upper bounds shown in Table 3.5 and the number of tokens of parallel data. For Spanish-English, the ratio is 0.35, for German-English it is 0.22, and for Bulgarian-English it is 0.18. This is only giving us very

	BG	**CS**	**DE**	**ES**	**FR**
Tokens (max)	19M	33M	176M	148M	174M
	JA	**KO**	**RU**	**UR**	
Tokens (max)	29M	22M	107M	1.9M	

Table 3.5: An upper bound for the amount of parallel data from Wikipedia for each language paired with English.

coarse estimates of the amount of parallel data available, but it gives us some idea of how much of an improvement we can expect when adding the mined data to an existing SMT system.

Chapter 4

Discriminative Sentence Alignment

In this chapter we will describe a discriminative models for performing sentence alignment on comparable document pairs. We use Wikipedia as our first source for comparable documents, and use a conditional random field (Lafferty et al., 2001) as the sentence alignment model. We also apply a discriminative monotonic alignment model to comparable documents mined from the Web as described by Uszkoreit et al. (2010).

4.1 Wikipedia as a Comparable Corpus

Wikipedia (Wikipedia, 2004) is an online collaborative encyclopedia available in a wide variety of languages. While the English Wikipedia is the largest, with over 3 million articles, there are 24 language editions with at least 100,000 articles.

French 496K	German 488K	Polish 384K	Italian 380K	Dutch 357K	Portuguese 323K	Spanish 311K	Japanese 252K
Russian 232K	Swedish 197K	Finnish 146K	Chinese 142K	Norwegian 141K	Volapük 106K	Catalan 103K	Czech 87K

Table 4.1: Number of aligned bilingual articles in Wikipedia by language (paired with English).

Articles on the same topic in different languages are also connected via "interwiki" links, which are annotated by users. This is an extremely valuable resource when extracting parallel sentences, as the document alignment is already provided. Table 4.1 shows how many of these "interwiki" links are present between the English Wikipedia and the 16 largest non-English Wikipedias.

Wikipedia's markup contains other useful indicators for parallel sentence extraction. The many hyperlinks found in articles have previously been used as a valuable source of information. (Adafre and de Rijke, 2006) use matching hyperlinks to identify similar sentences. Two links match if the articles they refer to are connected by an "interwiki" link. Also, images in Wikipedia are often stored in a central source across different languages; this allows identification of captions which may be parallel. Finally, there are other minor forms of markup which may be useful for finding similar content across languages, such as lists and section headings. In Section 4.2.3, we will explain how features are derived from this markup.

4.2 Models for Parallel Sentence Extraction

In this section, we will focus on methods for extracting parallel sentences from aligned, comparable documents. The related problem of automatic document alignment in news and web corpora has been explored by a number of researchers, including Resnik and Smith (2003), Munteanu and Marcu (2005), Tillmann (2009), and Tillmann and Xu (2009). Since our corpus already contains document alignments, we sidestep this problem, and will not discuss further details of this issue. That said, we believe that our methods will be effective in corpora without document alignments when combined with one of the aforementioned algorithms.

4.2.1 Binary Classifiers and Rankers

Much of the previous work involves building a binary classifier for sentence pairs to determine whether or not they are parallel (Munteanu and Marcu, 2005; Tillmann, 2009). The training data usually comes from a standard parallel corpus. There is a substantial class imbalance ($O(n)$ positive examples, and $O(n^2)$ negative examples), and various heuristics are used to mitigate this problem. Munteanu and Marcu (2005) filter out negative examples with high length difference or low word overlap (based on a bilingual dictionary).

We propose an alternative approach: we learn a ranking model, which, for each sentence in the source document, selects either a sentence in the target document that it is parallel to, or "null". This formulation of the problem avoids the class imbalance issue of the binary classifier.

In both the binary classifier approach and the ranking approach, we use a Maximum Entropy classifier, following Munteanu and Marcu (2005).

4.2.2 Sequence Models

In Wikipedia article pairs, it is common for parallel sentences to occur in clusters. A global sentence alignment model is able to capture this phenomenon. For both parallel and comparable corpora, global sentence alignments have been used, though the alignments were monotonic (Gale and Church, 1993; Moore, 2002; Zhao and Vogel, 2002). Our model is a first order linear chain conditional random field (CRF) (Lafferty et al., 2001). The set of source and target sentences are observed. For each source sentence, we have a hidden variable indicating the corresponding target sentence to which it is aligned (or null). The model is similar to the discriminative CRF-based word alignment model of (Blunsom and Cohn, 2006).

4.2.3 Features

Our features can be grouped into four categories.

4.2.3.1 Features derived from word alignments

We use a feature set inspired by (Munteanu and Marcu, 2005), who defined features primarily based on IBM Model 1 alignments (Brown et al., 1993). We also use HMM word alignments (Vogel et al., 1996) in both directions (source to target and target to source), and extract the following features based on these four alignments:[1]

1. Log probability of the alignment
2. Number of aligned/unaligned words
3. Longest aligned/unaligned sequence of words
4. Number of words with fertility 1, 2, and 3+

We also define two more features which are independent of word alignment models. One is a sentence length feature taken from (Moore, 2002), which models the length ratio between the source and target sentences with a Poisson distribution. The other feature is the difference in relative document position of the two sentences, capturing the idea that the aligned articles have a similar topic progression.

The above features are all defined on sentence pairs, and are included in the binary classifier and ranking model.

[1] These are all derived from the one best alignment, and normalized by sentence length.

4.2.3.2 Distortion features

In the sequence model, we use additional distortion features, which only look at the difference between the position of the previous and current aligned sentences. One set of features bins these distances; another looks at the absolute difference between the expected position (one after the previous aligned sentence) and the actual position.

4.2.3.3 Features derived from Wikipedia markup

Three features are derived from Wikipedia's markup. The first is the number of matching links in the sentence pair. The links are weighted by their inverse frequency in the document, so a link that appears often does not contribute much to this feature's value. The image feature fires whenever two sentences are captions of the same image, and the list feature fires when two sentences are both items in a list. These last two indicator features fire with a negative value when the feature matches on one sentence and not the other.

None of the above features fire on a null alignment, in either the ranker or CRF. There is also a bias feature for these two models, which fires on all non-null alignments.

4.2.3.4 Word-level induced lexicon features

In order to address sparsity issues in our seed parallel corpora, we introduce a bilingual lexicon model which learns word translation probabilities from the linked Wikipedia articles. The details of this model and the features derived from it can be found in (Smith et al., 2010).

4.3 Experiments

4.3.1 Data

We annotated twenty Wikipedia article pairs for three language pairs: Spanish-English, Bulgarian-English, and German-English. Each sentence in the source language was annotated with possible parallel sentences in the target language (the target language was English in all experiments). The pairs were annotated with a quality level: **1** if the sentences contained some parallel fragments, **2** if the sentences were mostly parallel with some missing words, and **3** if the sentences appeared to be direct translations. In all experiments, sentence pairs with quality **2** or **3** were taken as positive examples.

Language Pair	Binary Classifier			Ranker			CRF		
	Avg Prec	R@90	R@80	Avg Prec	R@90	R@80	Avg Prec	R@90	[illegible]
English-Bulgarian	75.7	33.9	56.2	76.3	38.8	57.0	**80.6**	**52.9**	[illegible]
English-Spanish	90.4	81.3	87.6	93.4	81.0	84.5	**94.7**	**87.6**	[illegible]
English-German	61.8	9.4	27.5	66.4	25.7	42.4	**78.9**	**52.2**	[illegible]

Table 4.2: Average precision, recall at 90% precision, and recall at 80% precision for each model in all three language pairs. In these experiments, the Wikipedia features and lexicon features are omitted.

For our seed parallel data, we used the Europarl corpus (Koehn, 2005) for Spanish and German and the JRC-Aquis corpus for Bulgarian, plus the article titles for parallel

Setting	Ranker			CRF		
	Avg Prec	R@90	R@80	Avg Prec	R@90	R@80
English-Bulgarian						
One Direction	76.3	38.8	57.0	80.6	52.9	59.5
Intersected	78.2	47.9	60.3	79.9	38.8	57.0
Intersected +Wiki	80.8	39.7	68.6	82.1	53.7	62.8
Intersected +Wiki +Lex	89.3	64.4	79.3	**90.9**	**72.0**	**81.8**
English-Spanish						
One Direction	93.4	81.0	84.5	94.7	87.6	90.2
Intersected	94.3	82.4	89.0	95.4	88.5	91.8
Intersected +Wiki	94.5	82.4	89.0	95.6	89.2	92.7
Intersected +Wiki +Lex	95.8	87.4	91.1	**96.4**	**90.4**	**93.7**
English-German						
One Direction	66.4	25.7	42.4	78.9	52.2	54.7
Intersected	71.9	36.2	43.8	80.9	54.0	67.0
Intersected +Wiki	74.0	38.8	45.3	82.4	56.9	**71.0**
Intersected +Wiki +Lex	78.7	46.4	59.1	**83.9**	**58.7**	68.8

Table 4.3: Average precision, recall at 90% precision, and recall at 80% precision for the Ranker and CRF in all three language pairs. "+Wiki" indicates that Wikipedia features were used, and "+Lex" means the lexicon features were used.

Wikipedia documents, and translations available from Wiktionary entries.[2]

4.3.2 Intrinsic Evaluation

Using 5-fold cross-validation on the 20 document pairs for each language condition, we compared the binary classifier, ranker, and CRF models for parallel sentence extraction. To tune for precision/recall, we used minimum Bayes risk decoding. We define the loss $L(\tau, \mu)$ of picking target sentence τ when the correct target sentence is μ as 0 if $\tau = \mu$, λ if $\tau = \text{NULL}$ and $\mu \neq \text{NULL}$, and 1 otherwise. By modifying the null

[2]Wiktionary is an online collaborative dictionary, similar to Wikipedia.

loss λ, the precision/recall trade-off can be adjusted. For the CRF model, we used posterior decoding to make the minimum risk decision rule tractable. As a summary measure of the performance of the models at different levels of recall we use average precision as defined in (Ido et al., 2006). We also report recall at precision of 90 and 80 percent. Table 4.2 compares the different models in all three language pairs.

In our next set of experiments, we looked at the effects of the Wikipedia specific features. Since the ranker and CRF are asymmetric models, we also experimented with running the models in both directions and combining their outputs by intersection. These results are shown in Table 4.3.

Identifying the agreement between two asymmetric models is a commonly exploited trick elsewhere in machine translation. It is mostly effective here as well, improving all cases except for the Bulgarian-English CRF where the regression is slight. More successful are the Wikipedia features, which provide an auxiliary signal of potential parallelism.

The gains from adding the lexicon-based features can be dramatic as in the case of Bulgarian (the CRF model average precision increased by nearly 9 points). The lower gains on Spanish and German may be due in part to the lack of language-specific training data. These results are very promising and motivate further exploration. We also note that this is perhaps the first successful practical application of an automatically induced word translation lexicon.

4.3.3 SMT Evaluation

We also present results in the context of a full machine translation system to evaluate the potential utility of this data. A standard phrasal SMT system (Koehn et al., 2003) serves as our testbed, using a conventional set of models: phrasal models of source given target and target given source; lexical weighting models in both directions, language model, word count, phrase count, distortion penalty, and a lexicalized reordering model. Given that the extracted Wikipedia data takes the standard form of parallel sentences, it would be easy to exploit this same data in a number of systems.

		German	English	Spanish	English	Bulgarian	E
Medium	sentences	924,416	924,416	957,884	957,884	413,514	41
	types	351,411	320,597	272,139	247,465	115,756	6
	tokens	11,556,988	11,751,138	18,229,085	17,184,070	10,207,565	10,42
Large	sentences	6,693,568	6,693,568	7,727,256	7,727,256	1,459,900	1,45
	types	1,050,832	875,041	1,024,793	952,161	239,076	13
	tokens	100,456,622	96,035,475	155,626,085	137,559,844	29,741,936	29,88
Wiki	sentences	1,694,595	1,694,595	1,914,978	1,914,978	146,465	14
	types	578,371	525,617	569,518	498,765	107,690	7
	tokens	21,991,377	23,290,765	29,859,332	28,270,223	1,455,458	1,51

Table 4.4: Statistics of the training data size in all three language pairs.

For each language pair we explored two training conditions. The "Medium" data condition used easily downloadable corpora: Europarl for German-English and Spanish-English, and JRC/Acquis for Bulgarian-English. Additionally we included titles of all linked Wikipedia articles as parallel sentences in the medium data con-

		German	English	Spanish	English	Bulgarian	English
Dev A	sentences	2,000	2,000	2,000	2,000	2,000	2,000
	tokens	16,367	16,903	24,571	21,493	39,796	40,503
Test A	sentences	5,000	5,000	5,000	5,000	2,473	2,473
	tokens	42,766	43,929	68,036	60,380	52,370	52,343
Wikitest	sentences	500	500	500	500	516	516
	tokens	8,235	9,176	10,446	9,701	7,300	7,701

Table 4.5: Statistics of the test data sets.

dition. The "Large" data condition includes all the medium data, and also includes using a broad range of available sources such as data scraped from the web (Resnik and Smith, 2003), data from the United Nations, phrase books, software documentation, and more.

In each condition, we explored the impact of including additional parallel sentences automatically extracted from Wikipedia in the system training data. For German-English and Spanish-English, we extracted data with the null loss adjusted to achieve an estimated precision of 95 percent, and for English-Bulgarian a precision of 90 percent. Table 4.4 summarizes the characteristics of these data sets. We were pleasantly surprised at the amount of parallel sentences extracted from such a varied comparable corpus. Apparently the average Wikipedia article contains at least a handful of parallel sentences, suggesting this is a very fertile ground for training MT systems.

The extracted Wikipedia data is likely to make the greatest impact on broad domain test sets – indeed, initial experimentation showed little BLEU gain on in-domain test sets such as Europarl, where out-of-domain training data is unlikely to provide appropriate phrasal translations. Therefore, we experimented with two broad domain test sets.

First, Bing Translator provided a sample of translation requests along with translations in German-English and Spanish-English – this constituted our standard development and test set for those language pairs. Unfortunately no such tagged set was available in Bulgarian-English, so we held out a portion of the large system's training data to use for development and test. In each language pair, the test set was split into a development portion ("Dev A") used for minimum error rate training (Och, 2003) and a test set ("Test A") used for final evaluation.

Language pair	Training data	Dev A	Test A	Wikitest
Spanish-English	Medium	32.6	30.5	33.0
	Medium+Wiki	36.7 (+4.1)	33.8 (+3.3)	39.1 (+6.1)
	Large	39.2	**37.4**	38.9
	Large+Wiki	**39.5** (+0.3)	37.3 (-0.1)	**41.1** (+2.2)
German-English	Medium	28.7	26.6	13.0
	Medium+Wiki	31.5 (+2.8)	29.6 (+3.0)	18.2 (+5.2)
	Large	**35.0**	33.7	17.1
	Large+Wiki	34.8 (-0.2)	**33.9** (+0.2)	**20.2** (+3.1)
Bulgarian-English	Medium	36.9	26.0	27.8
	Medium+Wiki	37.9 (+1.0)	27.6 (+1.6)	37.9 (+10.1)
	Large	**51.7**	**49.6**	36.0
	Large+Wiki	**51.7**(+0.0)	49.4 (-0.2)	**39.5**(+3.5)

Table 4.6: BLEU scores of MT systems under various training and test conditions. The final BLEU score from minimum error rate training is in the first column; two additional columns are BLEU scores on held-out test sets. For training data conditions including the extracted Wikipedia sentences, the parenthesized values indicate the absolute BLEU difference against the corresponding system without Wikipedia extracts.

Second, we created new test sets in each of the three language pairs by sampling parallel sentences from held out Wikipedia articles. To ensure that this test data was clean, we manually filtered the sentence pairs that were not truly parallel and edited them as necessary to improve adequacy. We called this "Wikitest". Characteristics of these test sets are summarized in Table 4.5.

We evaluated the resulting systems using BLEU-4 (Papineni et al., 2002); the results are presented in Table 4.6. First we note that the extracted Wikipedia data are very helpful in medium data conditions, significantly improving translation performance in all conditions. Furthermore we found that the extracted Wikipedia sentences substantially improved translation quality on held-out Wikipedia articles. In every case, training on medium data plus Wikipedia extracts led to equal or better translation quality than the large system alone. Furthermore, adding the Wikipedia data to the large data condition still made substantial improvements.

4.4 Comparable documents from the Web

Multilingual websites have often been used as a source of parallel data (Resnik and Smith, 2003; Huang et al., 2005; Shi et al., 2006). Most approaches for finding potential parallel documents rely on metadata rather than the content of the websites. This metadata may include links which appear to point to alternate versions of the same page in another language, the URLs of the websites, or their HTML structure. Another approach for finding multilingual websites is given by Uszkoreit et al. (2010), who translate all non-English web pages into English using their MT system and then use monolingual document similarity measures to find comparable document pairs. Uszkoreit et al. (2010) also describe a sentence alignment model which is applied to the comparable document pairs. In this work, we will instead use a discriminative, monotonic alignment model to align these document pairs. As this requires labeled sentence alignment data for both training and evaluation, we also describe an annotation tool for monotonic alignments which allows $m : n$ sentence matchings.

4.4.1 Data collection

4.4.1.1 Annotation Tool

We obtain a set of Japanese-English document pairs using the method described by Uszkoreit et al. (2010). From this set, we randomly selected roughly 1000 document pairs to be annotated. The annotators were presented with the interface shown in Figure 4.1. For each document pair we asked if the two documents are in the correct language pair, and if one of the documents appeared to be machine translated. If the language pair was correct and the documents appeared to be translated by a human, then the annotators aligned the sentences in the document.

The annotation tool is closely related to the monotonic alignment model in how

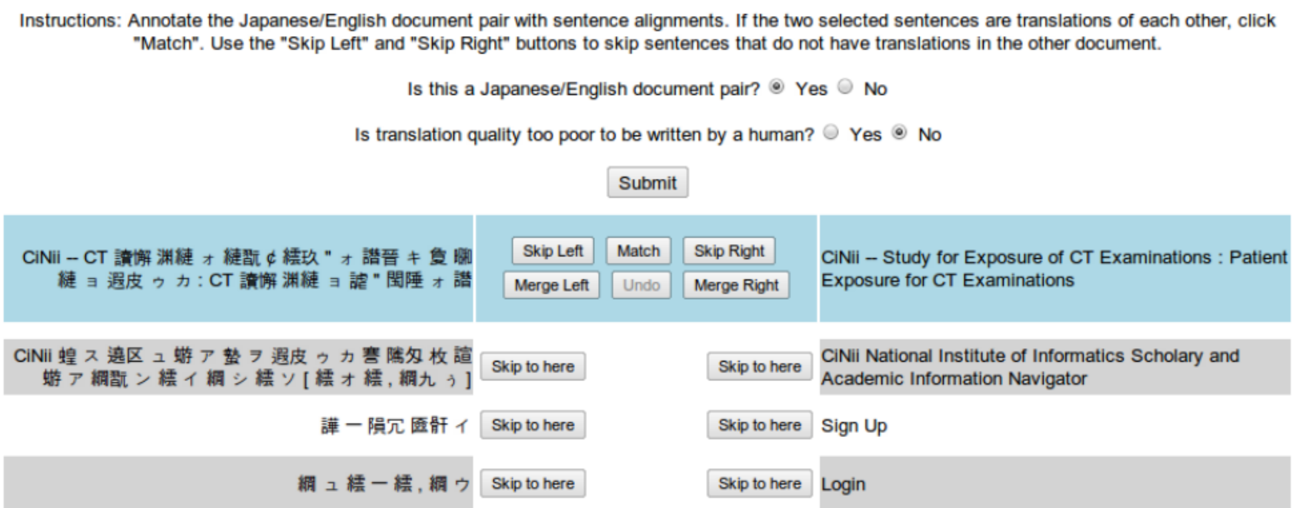

Figure 4.1: The annotation interface for sentence alignment. **[Take or adapt the figure from the Google slides. –JS]**

it operates. The user is asked whether or not the first sentences in each document are parallel. If they are, the user selects "Match", the sentences are aligned, and the user is then prompted about the next two sentences. If the sentence pair presented to the annotator is not parallel, they will select either "Skip Left" or "Skip Right", which advance the current sentence in the document on the left or right. These three operations correspond to the operations used in the monotonic alignment model, and the annotator's sequence of operations can be directly used as training data.

The tool also allows the annotator to specify arbitrary $n : m$ sentence alignments throught the "Merge Left" and "Merge Right" buttons. The merge actions append the immediately following sentences to the currently selected sentences, which can then be aligned.

4.4.1.2 Data Collection Results

Since the data we have annotated using this tool is taken from the HTML source of webpages, and converted to plain text and segmented into sentences automatically, the "Merge" buttons are often used to correct errors in sentence segmentation. This results in a few large $n : m$ alignments, though most alignments are still $1 : 1$. Table 4.7 breaks down the alignments that the annotators have provided.

Alignment Type	1:1	2:1	1:2	2:2	Other
Percentage	95.8%	1.70%	1.28%	0.159%	1.12%

Table 4.7: Statistics on the types of $n : m$ alignments found in the annotated data.

4.4.2 Monotonic alignment model

Our sentence alignment model is a discriminatively trained monotonic alignment model. An illustration of this model is given in Figure 4.2. Formally, we represent the model as a weighted finite-state automata (WFSA) with features that fire on each arc as described in Eisner (2002). Specifically, in this work we use a weighted finite-state transducer (WFST) to describe our alignment model. A WFST includes a set of states $\mathbb{Q}$, source and target alphabets Σ_S and Σ_T, transition function $\delta : \mathbb{Q} \times \Sigma_S \times \Sigma_T \longrightarrow \mathbb{Q} \times W$, and start and end states $\mathbb{S}$ and $\mathbb{F}$.

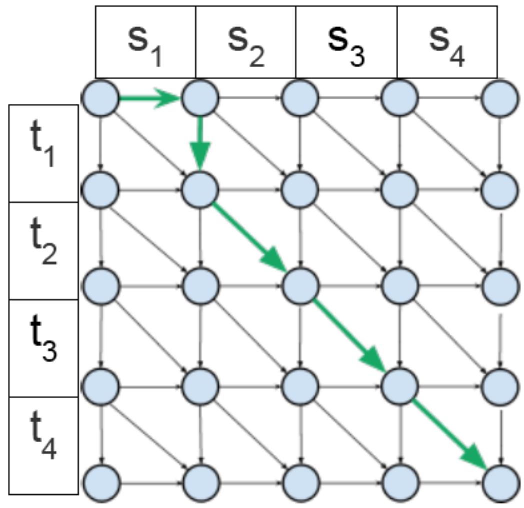

Figure 4.2: The discriminative model for monotonic sentence alignment. The highlighted path represents a possible alignment between the two documents.

$$
\begin{aligned}
\mathbb{Q} &= \{q_{i,j} \mid 0 \leq i \leq |\vec{S}|, 0 \leq j \leq |\vec{T}|\} \\
\Sigma_S &= \{\epsilon\} \cup \{S_i \mid 0 \leq i < |\vec{S}|\} \\
\Sigma_T &= \{\epsilon\} \cup \{T_j \mid 0 \leq j < |\vec{T}|\} \\
\delta(q_{i,j}) &= \{(q_{i+1,j}, (S_i, \epsilon), \vec{f}(S_i, \epsilon) \cdot \vec{w}), \\
&\quad (q_{i,j+1}, (\epsilon, T_j), \vec{f}(\epsilon, T_j) \cdot \vec{w}), \\
&\quad (q_{i+1,j+1}, (S_i, T_j), \vec{f}(S_i, T_j) \cdot \vec{w})\} \\
\mathbb{S} &= q_{0,0} \\
\mathbb{F} &= q_{|\vec{S}|,|\vec{T}|}
\end{aligned}
$$

[The transition function still needs some work, and I need to mention semirings. A section on notation should help with

this. –JS]

The FSM representing the set of possible alignments for a document pair $(\vec{S}, \vec{T})$ has $(|\vec{S}|+1) \cdot (|\vec{T}|+1)$ states. These states correspond to positions in the source and target documents. Each state has at most three transitions, which either skip the current source sentence, skip the current target sentence, or align the current source and target sentences. Note that these operations directly match the actions used by the annotators to generate the labeled alignments, so they may be directly used as an observed path through the WFSA. This automata only contains $1:1$ alignments, though $n:m$ alignments could be added to this machine through additional arcs on each state.

The weights on each of the arcs a come from the dot product of the features which fire on that arc and the weight vector $\vec{w}$. The total weight of a path $\pi = a_0, a_1, \ldots, a_n$ is the dot product of all features firing on each arc and the weight vector. Using this property we can now define a probability distribution over paths through this WFST:

$$p(\pi|\vec{S}, \vec{T}) = \frac{\sum_{a \in \pi} \exp \vec{f}(a) \cdot \vec{w}}{\sum_{\pi'} \sum_{a \in \pi'} \exp \vec{f}(a) \cdot \vec{w}}$$

The denominator is the sum of the weights of all paths through the WFST. We train our model to maximize the probability of the observed training data:

$$\underset{\vec{w}}{\operatorname{argmax}} \sum_{(\pi, \vec{S}, \vec{T})} p(\pi|\vec{S}, \vec{T})$$

The gradient of this objective can be computed using standard finite-state algorithms (Eisner, 2002).

4.4.2.1 Features

We chose to use a simple set of features which are mostly based on bag-of-words similarity measures. First, we have bias features which fire on all "matching" and "mismatching" arcs (arcs which emit (S_i, T_j) are matching arcs, and the mismatching arcs emit either (ϵ, T_j) or (S_i, ϵ)). For the matching arcs, we project the source sentence through a weighted bilingual dictionary and compute cosine similarity with the target sentence (and the same is done in reverse).**[I don't have the exact set of features used in the experiments from my Google slides. –JS]**

We also experimented with a set of first-order features, which look at the previous arc taken in the WFST. This is made possible by splitting each state $q_{i,j}$ into $q_{i,j,+}$ and $q_{i,j,-}$. "Match" transitions lead to $q_{i,j,+}$ while "mismatch" transitions lead to $q_{i,j,-}$. Since the states are now recording the last operation, the arcs leaving these states have features which fire on two matches in a row, for example. The intuition behind adding these features is that matches often follow other matches, and a similar trend is present for mismatches.

Model	Precision	Recall	F1
Baseline	63.9%	95.0%	76.4%
Discriminative Aligner	**66.0%**	94.4%	**77.7%**
+First Order Features	65.1%	**95.2%**	77.3%

Table 4.8: Precision, recall and F1 score measured on a held out set of aligned Japanese-English document pairs.

4.4.3 Results

We compare our discriminatively trained model against a hand-tuned monotonic alignment model. Our results are shown in Table 4.8.

Chapter 5

Comparable Data on Twitter

Chapter 6

Unsupervised Parallel Sentence Extraction from Comparable Corpora

6.1 Unsupervised Sentence Alignment

In most previous work on finding parallel sentences in comparable corpora, some initial parallel data (parallel sentences or bilingual dictionary entries) is used as a starting point. This data is used to extract parallel sentences, with the hope that the bilingual word correspondences from the initial data are enough to determine whether or not two sentences are parallel. The obvious drawback is the reliance on the initial data, which may be small. Ideally, one would learn additional word correspondences from parallel sentences that were extracted, and this information could be used to find more parallel sentences. In fact, this bootstrapping method has been used in previous work (Fung and Cheung, 2004a,b; Wu and Fung, 2005).

We will explore a novel way of using semi-supervised learning to find parallel sentences: by including sentence and word alignment in a single model. Much like the IBM word alignment models (Brown et al., 1993) which can be trained on sentence pairs without word alignment data, our model can be trained on document pairs without sentence or word alignment data, and can similarly be trained using the expectation-maximization (EM) algorithm (Dempster et al., 1977).

6.1.1 Model

First we must define a generative model of a bilingual (possibly) parallel document pair. We will use a joint model of the source and target documents based on stochastic edit distance (Ristad and Yianilos, 1998). Document pairs are generated by a memoryless transducer which generates substitution pairs (S, T), insertion pairs (ϵ, T), deletion pairs (S, ϵ), and the termination pair (ϵ, ϵ), borrowing the convention used by (Oncina and Sebban, 2006) for simplicity. Substitution pairs correspond to parallel source and target sentences, while the insertion and deletion pairs are monolingually generated. For this model to be properly defined, the probability of generating all pairs must sum to one:

$$\sum_{x \in S \cup \{\epsilon\}, y \in T \cup \{\epsilon\}} p(x, y) = 1 \tag{6.1}$$

Since the insertion and deletion operations are monolingual generation of sentences, we use a standard n-gram language model for their probabilities. For the probability of a substitution pair, we decompose $p(S,T)$ into $p(T|S)p(S)$. $p(T|S)$ is defined by an IBM word alignment model (Brown et al., 1993) (Model 1 in this preliminary work), and $p(S)$ is given by the same language model used to generate deletion pairs $((S,\epsilon))$. Since $p(S,T)$, $p(S,\epsilon)$ and $p(\epsilon,T)$ all individually sum to one, they must be weighted to ensure that $p(\mathbf{S},\mathbf{T})$ is properly normalized.[1] In this work, we will use a single parameter to weight these pairs:

$$\begin{aligned} p(S,T) &= \lambda p_{Model1}(T|S)p_{LM}(S) \\ p(S,\epsilon) &= \frac{1-\lambda}{2} p_{LM}(S) \\ p(\epsilon,T) &= \frac{1-\lambda}{2} p_{LM}(T) \end{aligned}$$

p_{Model1} and p_{LM} refer to the IBM Model 1 and a unigram language model, respectively. The parameter λ roughly controls how eager the model is to label sentence pairs as parallel. This can be set based on some prior knowledge about the corpus. p_{Model1} is given by the following equation from (Brown et al., 1993):

[1] Since our document pairs are always observed, we can safely ignore the stopping cost $p(\epsilon,\epsilon)$ by assuming it to be some small constant.

$$p(T|S) = p\left(|T|\big||S|\right) \frac{1}{|S|^{|T|}} \prod_{j=1}^{|T|} \sum_{i=1}^{|S|} p(t_j|s_i) \tag{6.2}$$

For simplicity, we assume the source sentence S contains the null word. The term $\frac{1}{|S|^{|T|}}$ is the uniform alignment probability. The length distribution, $p\left(|T|\big||S|\right)$, was originally described as a uniform distribution over a large finite set of lengths. Since Model 1 is usually applied to parallel corpora with observed sentence alignments, and the goal of using Model 1 is to find word translation probabilities ($p(t|s)$), it is unnecessary to find an accurate model of sentence length. However, when the sentence alignments are being learned, it is important to have an accurate model of the length of the target sentence given the source sentence. In this work, we use a Poisson distribution to model the target sentence length, following Moore (2002).

The probability for generating sentences monolingually, $p_{LM}(S)$, is a unigram model estimated from the source language documents in the corpus. Similarly, $p_{LM}(T)$ is estimated form the target language documents. While a higher order language model could be learned, we use a unigram model to more closely match IBM Model 1, which can be thought of as a mixture of unigram models (one for each source word and one for the null word) that generate the target sentence. We also use a Poisson distribution to model the lengths of monolingually generated sentences, rather than generating a special end-of-sentence token.

6.2 Data Collection

In order to evaluate the unsupervised sentence alignment model that we are proposing, we must have bilingual document pairs with an annotated sentence alignment. While existing parallel corpora may be used for this, the document pairs in these corpora are highly parallel and would not resemble the alignments found in Wikipedia articles on the same topic, or comparable news articles. We will instead annotate comparable document pairs with their sentence alignment using Amazon's Mechanical Turk (MTurk).

6.2.1 Mechanical Turk

MTurk is an online marketplace where people may post collections of tasks that workers may choose to complete for small amounts of money. These tasks are referred to as Human Intelligence Tasks (or HITs) because they are intended to be easy for humans to complete but difficult to automate. Examples of HITs include the identification of offensive images, moderation of forum posts or blog comments, and finding the contact information of a business. The workers on MTurk are referred to as "Turkers". MTurk has also been used for several natural language tasks (Snow et al., 2008), including the evaluation of machine translation output (Callison-Burch, 2009) and even translation itself (Zaidan and Callison-Burch, 2011). The greatest concern when using MTurk for annotation is ensuring that the results are reliable.

There are many ways in which sentence alignment of bilingual comparable documents could be organized into HITs on MTurk. The simplest way would be to take all possible sentence pairs in the document pair, and ask the Turkers to decide whether or not they are parallel. Unfortunately, this will result in far too many tasks to be affordable, as some Wikipedia articles have over a thousand sentences. In order to cut down on the number of tasks, we applied pruning to the candidate sentence pairs.

6.2.2 Pruning and Data Selection

Our pruning strategy is roughly based on that of Munteanu and Marcu (2005). Sentence pairs are filtered by two criteria. **Length ratio:** The ratio between the lengths (in words) of the two sentences must be below a threshold in each direction. **Coverage:** The percentage of target words t which either have an exact string match with a source word, or have $p(t|s)$ (under IBM Model 1) greater than a threshold for some s in the source sentence. We obtain the Model 1 probabilities by training on existing parallel data and bilingual dictionary entries for the language pair. Coverage is computed on both the source and target sentences, and a sentence pair is filtered if the average coverage falls below a threshold.

This pruning strategy requires three thresholds to be set: a maximum length ratio, a minimum average source/target coverage, and a minimum Model 1 probability for determining whether or not a word is covered. We tune these thresholds on existing parallel data to ensure that the filter has high recall (90%) while still removing many non-parallel sentence pairs. For our Urdu/English experiments, the thresholds we used were 2.5 for the maximum length ratio, 0.01 for the minimum average coverage, and 0.575 for the Model 1 word coverage threshold. We take our parallel data for training Model 1 parameters from the NIST MT09 Urdu-English training set and the bilingual dictionaries and sentences gathered by Post et al. (2012).

In addition to pruning sentence pairs which are not likely parallel, we also remove any pairs containing sentences with less than five tokens. Wikipedia articles include section headings lists of names (such as an actor's filmography), and links to other articles or external websites. Since our goal is to find parallel sentences, we do not ask Turkers to annotate these very short segments.

Since we are not asking Turkers to annotate all possible sentence pairs from an article pair, evaluation becomes more difficult. We will discuss how we use our partial annotation in Section 6.2.5.

6.2.3 Task Design

Our strategy for designing the HITs on MTurk was to give the user an Urdu sentence and a list of up to ten English sentences. The Turker is asked to select which of the English sentences is parallel to the Urdu sentence, or select "None of the above" if none of the English sentences are parallel. We also ask if the sentence pair they find is a partial or full match, and give some examples of each in the instructions. Figure 6.1 shows an example of one of these questions.

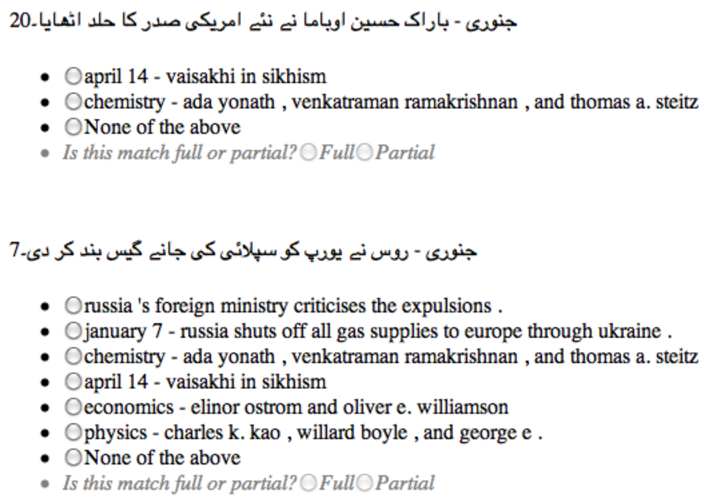

Figure 6.1: The MTurk annotation interface for finding Urdu-English parallel sentences.

Our method of pruning potential sentence pairs may leave us with more than ten candidate English sentences for some Urdu sentences. When this happens, we make additional questions about these Urdu sentences to ensure all candidate pairs are accounted for in the annotation.

In each HIT, we ask the Turkers to annotate up to ten Urdu sentences with their English counterpart (if any), including two control questions with sentences taken from the parallel data described in Section 6.2.2. There is one positive and one negative control in each HIT. We also request that each HIT be done by three Turkers.

6.2.4 Data Collection Results

In our first large-scale experiment, we took 92 Urdu-English article pairs, applied our filters as described in Section 6.2.2, and uploaded our task to MTurk. While there were over 8 million possible sentence pairs in these articles before pruning, we ended up with $785,000$ sentence pairs to be annotated at a total cost of \$726.80 (this cost includes the duplicate annotations).

Agreement among the Turkers was high ($\kappa = 0.84$). While the most common answer was "None of the above", there were a substantial number of Urdu sentences which the Turkers found some English counterpart for. For 21.4% of Urdu sentences, at least one Turker found one of the English sentences to be parallel, and in 44.8% of Urdu sentences, at least two Turkers identified a match.

6.2.5 Evaluation Using Partial Alignments

When we evaluate our sentence pair alignment model, we would like to compute the precision and recall of the proposed sentence alignments. However, since we prune many possible sentence pairs before asking the Turkers for annotation, we cannot be sure whether or not some sentence pairs are parallel. In this section, we will outline a scheme for evaluating sentence alignments using our partially annotated data.

Our primary intrinsic evalutaion metric is alignment F-measure on sentence alignments. This metric could also be seen as F-measure on a parallel sentence pair retrieval task. Let T be the set of true positives (sentence pairs that are truly parallel), and P be the set of predicted positives (sentence pairs identified by our model as parallel). Precision, recall, and F-measure are defined as follows:

$$\begin{aligned}
\text{Precison} &= \frac{|T \cup P|}{|P|} \\
\text{Recall} &= \frac{|T \cup P|}{|T|} \\
\text{F-measure} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}$$

When our document pairs are only partially annotated, we will used modified definitions of precision, recall, and F-measure. Let U be the set of sentence pairs which were not annotated as parallel or non-parallel.

$$\begin{aligned}
\text{Precison} &= \frac{|T \cup P|}{|P \setminus U|} \\
\text{Recall} &= \frac{|T \cup P|}{|T|} \\
\text{F-measure} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}$$

Since T and U are disjoint, only the definition of precision needs to be modified.

Given the annotations we gathered from MTurk, it is possible to define U in multiple ways. The most conservative method would be to take U to be all sentence pairs not presented to the Turkers. However, if we make the assumption that sentence alignments of the document pairs are $1:1$, then when a Turker annotates a sentence

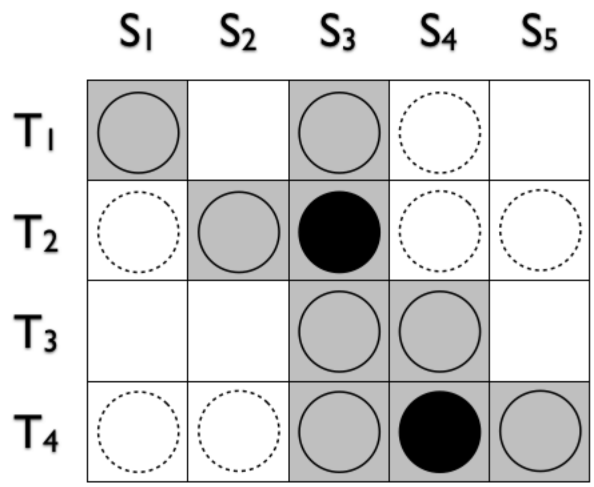

Figure 6.2: A partial alignment grid for a comparable document pair. The shaded cells in the grid represent the sentence pairs which were presented to the Turkers for annotation. A filled circle indicates the Turker found the sentence pair to be parallel, and an empty circle means the pair is not parallel. The dashed circles represent the sentence pairs we infer to be non-parallel by assuming the sentence alignments are 1 : 1.

pair (S, T) as parallel, it follows that all (S, T') pairs with $T' \neq T$ and (S', T) pairs with $S' \neq S$ are not parallel. Since the alignments we found were mostly 1 : 1, we decided to go with this option.[2] Figure 6.2 illustrates this method.

6.2.6 Alternate Evaluation Strategies

Section 6.2.5 describes a method for using Turkers' partial annotation of a document pair's sentence alignment for intrinsic evaluation of a sentence alignment model. In this section, we will explore other strategies for using the Turkers' output for evaluation.

[2]There were a small number of alignments which were not 1 : 1, most of which were image captions.

In our MTurk task setup (see Section 6.2.3), we collect redundant annotations for each HIT. While this was done primarily for quality control, and it is more convenient to use a single judgement for each sentence pair, we can perform a more fine-grained analysis by looking at the individual Turkers' judgements. Also, we gave the option of labeling a parallel sentence pair as a "partial" or "full" match.

[TODO: For the semi-supervised experiments, we want to treat sentence pairs that any annotator marked as a match as a true positive. We could also measure the inter-annotator agreement of our system against the Turkers. –JS]

6.3 Experiments

Our first set of experiments uses a semi-supervised setting. We have both parallel sentences (labeled data) and comparable document pairs (unlabeled data), and learn our model's parameters from both of these resources.

Our parallel corpus is taken from the NIST MT09 Urdu-English training set and the bilingual dictionaries and sentences gathered by Post et al. (2012).[3] The parallel sentences from this corpus are treated as single sentence document pairs. Alternatively, the entire training set could be seen as a single document pair whose sentence alignment lies completely on the diagonal. The model described in 6.1 does not dif-

[3]This is the same parallel corpus used to create the sentence pair filters used in collecting the annotated sentence alignments.

ferentiate between these two ways of viewing the corpus. In either case, learning from the parallel sentences is identical to IBM Model 1 training.

The comparable document pairs are a subset of the Wikipedia article pairs that we annotated using MTurk as described in Section 6.2. 60% of this data was taken as a development set. The remaining 40% of the annotated document pairs was split into two equal sized test sets.[4]

In the following experiments the setup is as follows: We initialize our parameters by running five iterations of EM on the parallel sentences from our labeled data. Then we run several iterations of EM on both the labeled data and unlabeled data, measuring performance after each iteration.

6.4 Experiments (Alternate)

In this section, we explore the relationship between the amount of initial parallel data, the quality of the extracted parallel data, and the end-to-end machine translation quality. We start with Spanish-English as our language pair, since this is a high resource language pair, and we can always simulate a low resource setting by restricting the amount of data used.

[4]This split was done in order to have training, development, and test sets for supervised sentence aligment models.

6.4.1 Datasets

For our initial parallel data, we use the parallel and monolingual corpora available for the 2010 Machine Translation Workshop's shared task (WMT10). For the Spanish-English task, the WMT10 data includes Europarl version 5 (we use version 6 in our experiments**[This may change to 7 –JS]**) (Koehn, 2005), the United Nations parallel text, and parallel and monolingual news corpora. Table 6.1 lists the corpora used in detail.

		Spanish	English	English (Monolingual)
Europarl	Sentences	1.79M	1.79M	1.79M
	Tokens	46.8M	44.7M	44.7M
United Nations	Sentences	6.22M	6.22M	6.22M
	Tokens	191M	164M	164M
News Commentary	Sentences	98.6K	98.6K	126K
	Tokens	2.45M	2.10M	261M
News	Sentences	N/A	N/A	48.7M
	Tokens	N/A	N/A	989M

Table 6.1: Statistics for the initial parallel/monolingual data used in training baseline MT systems and for extracting new parallel data. The monolingual data is only used for language modeling, not for extracting parallel sentences.

This data is used both for training the parallel sentence extractor, and as the initial data in the MT system.

6.4.2 Supervised Parallel Sentence Extraction

In order to extract parallel sentence pairs from Wikipedia, we used a simplified version of the approach described in Smith et al. (2010).

Using the initial parallel data and a small amount of annotated Spanish-English Wikipedia articles, we extracted sentence pairs from all of the Spanish-English Wikipedia articles which were identified as sharing a topic through Wikipedia's Interwiki link system. This gave us a set of 433 thousand comparable document pairs. For all pairs of sentences in each document pair, we applied a binary classifier to determine whether or not the sentence pair was parallel.

Table 6.2 lists the parallel corpora extracted from Spanish-English article pairs from Wikipedia. "Wiki@X" refers to the parallel sentences extracted with a classification threshold of X (a lower classification threshold will allow more sentences to be extracted). The monolingual data was taken from the English side of all Spanish-English document pairs, making it consistent across conditions.

		Spanish	English	English (Monolingual)
Wiki@0.75	Sentences	989K	989K	14.8M
	Tokens	32.0M	37.1M	286M
Wiki@0.5	Sentences	1.60M	1.60M	14.8M
	Tokens	44.8M	51.0M	286M
Wiki@0.25	Sentences	2.38M	2.38M	14.8M
	Tokens	70.4M	79.6M	286M

Table 6.2: Statistics for parallel corpora extracted from Wikipedia.

6.4.3 Results

We report end-to-end MT results using the initial parallel data and the extracted parallel data from Wikipedia. For our baseline MT system we used the phrase-based model included in the Moses toolkit Koehn et al. (2007) with all options set to the default.

We used two test sets to evaluate the end-to-end MT performance: the test set from WMT10 which was taken from the news domain, and a set of parallel sentences from Wikipedia gathered by (Smith et al., 2010).

	WMT10	Wikipedia
Europarl Only	24.75	—
+Wiki LM	26.91	—
+Wiki Parallel (@0.75)	27.22	—
+Wiki Parallel (@0.5)	27.41	—
+Wiki Parallel (@0.25)	—	—
All Initial Corpora	28.51	—
+Wiki LM	28.55	—
+Wiki Parallel (@0.75)	—	—
+Wiki Parallel (@0.5)	29.23	—
+Wiki Parallel (@0.25)	—	—

Table 6.3: BLEU scores for systems trained on different sets of parallel and monolingual data before and after adding data from Wikipedia.

6.5 Supervised Parallel Sentence Extraction with Low Resources

In most previous work, a large amount of existing parallel data is used to find new parallel sentence pairs in comparable corpora. Here, we explore using parallel sentence extraction methods which use far less initial data. Specifically, we will only be using information available in Wikipedia: Wiktionary translations, the titles of bilingually linked articles, and the text of these articles as sources of initial parallel data. We make use of this limited data in several ways:

1. Treat the Wiktionary entries and article titles as regular parallel data and learn the standard word alignment features from this data.

2. Use the data as a bilingual dictionary to project source language sentences into the target language (and vice-versa), and use vector space similarity measures to compare the source/target sentences.

3. Treat punctuation and numeric characters specially by not projecting them through the dictionary. This can be done when large amounts of initial data are available, but there is less need to.

In addition, we explore different ways of augmenting the initial data:

1. Give each source/target token a vector representation: a bit vector with dimensionality equal to the number of comparable document pairs. Each bit indicates whether or not the token appears in the comparable document pair. This representation places source and target tokens in the same space, allowing us to identify words which are translations of one another (see Fung and Church (1994) for a similar approach).

2. Create bilingual dictionary entries for the anchor text of hyperlinks within Wikipedia and the foreign title of the article they link to.

3. Use Wiktionary's morphological data to lemmatize tokens of source/target text.

6.5.1 Intrinsic Evaluation

We use the annotated comparable documents gathered by Smith et al. (2010) as a source of training and test data. In addition, we use parallel data as a source of annotated sentence pairs, since annotated comparable data is not available for many language pairs. We use this data to create three experimental conditions:

1. Train on parallel data, test on parallel data.
2. Train on parallel data, test on comparable data.
3. Train on comparable data, test on comparable data.

Ultimately, the classifiers are used on comparable data, so the relationship between performance on parallel and comparable data will be explored here.

6.5.2 Extrinsic Evaluation

We would also like to evaluate performance for languages without any initial parallel data. In this scenario, there is no training or test data for the sentence pair classifier, so we must either take feature values from classifiers trained on other language pairs, or use a single feature and threshold this feature's score to extract sentence pairs.

6.5.3 Generalizing Classifiers Across Languages

The sentence pair classifiers here use a small set of dense features which are independent of the language pair, though computation of the feature values uses data that is language pair dependent. It is possible to take feature weights trained for one language pair and apply them to another language pair. This is desirable because annotated comparable data is only available for a small number of language pairs, and while parallel data can be used as a substitute, it does not match the test condition as well. In this section, we explore the viability of transferring feature weights across language pairs where we do have access to annotated comparable data. First, we will examine the two main sources of variability: differences in training data, and differences in feature values.

6.5.3.1 Feature Values

The majority of the features used in the sentence pair classifier make use of existing bilingual data in the form of parallel sentences or bilingual dictionaries. The amount of data available will vary by language pair, and even with identical amounts of data, the distribution of feature values can still vary greatly depending on factors such as the morphological complexity of the languages involved. Figure **??** gives the distribution of the coverage feature in Spanish-English and German-English. For languages with larger vocabularies, the scores tend to be lower.

6.6 Conclusions

(Modest gains with large amounts of existing parallel data, ??? gains with little existing data)

Bibliography

Sisay F. Adafre and Maarten de Rijke. Finding Similar Sentences Across Multiple Languages in Wikipedia. In EACL 2006 Workshop on New Text, Wikis and Blogs and Other Dynamic Text Sources, 2006.

Satanjeev Banerjee and Alon Lavie. 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, 2005.

Phil Blunsom and Trevor Cohn. Discriminative word alignment with conditional random fields. In Proceedings of ACL, 2006.

Peter Brown, John Cocke, S Della Pietra, V Della Pietra, Frederick Jelinek, Robert Mercer, and Paul Roossin. A statistical approach to language translation. In Proceedings of the 12th conference on Computational linguistics-Volume 1, pages 71–76. Association for Computational Linguistics, 1988.

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick

Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. Computational linguistics, 16(2):79–85, 1990.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. Aligning sentences in parallel corpora. In Proceedings of the 29th annual meeting on Association for Computational Linguistics, ACL '91, pages 169–176, Stroudsburg, PA, USA, 1991. Association for Computational Linguistics. doi: 10.3115/981344.981366. URL http://dx.doi.org/10.3115/981344.981366.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. Comput. Linguist., 1993.

Chris Callison-Burch. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1, EMNLP '09, pages 286–295, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL http://dl.acm.org/citation.cfm?id=1699510.1699548.

Jiang Chen and Jian-Yun Nie. Parallel web text mining for cross-language ir. In IN IN PROC. OF RIAO, pages 62–77, 2000.

Stanley F. Chen. Aligning sentences in bilingual corpora using lexical information. In Proceedings of the 31st annual meeting on Association for Computational

Linguistics, ACL '93, pages 9–16, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/981574.981576. URL http://dx.doi.org/10.3115/981574.981576.

Kenneth Ward Church. Char align: a program for aligning parallel texts at the character level. In Proceedings of the 31st annual meeting on Association for Computational Linguistics, ACL '93, pages 1–8, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics. doi: 10.3115/981574.981575. URL http://dx.doi.org/10.3115/981574.981575.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: Rational, evaluation and approaches. Natural Language Engineering, 16 (01):105–105, 2010.

J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation-Volume 6, pages 10–10. USENIX Association, 2004.

A.P. Dempster, N.M. Laird, D.B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38, 1977.

Andreas Eisele and Yu Chen. Multiun: A multilingual corpus from united nation documents. In Daniel Tapias, Mike Rosner, Stelios Piperidis, Jan Odjik, Joseph Mariani, Bente Maegaard, Khalid Choukri, and Nicoletta Calzolari (Conference

Chair), editors, Proceedings of the Seventh conference on International Language Resources and Evaluation, pages 2868–2872. European Language Resources Association (ELRA), 5 2010.

Jason Eisner. Parameter estimation for probabilistic finite-state transducers. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073085. URL http://dx.doi.org/10.3115/1073083.1073085.

Pascale Fung and Percy Cheung. Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In Proceedings of the 20th international conference on Computational Linguistics, COLING '04, Stroudsburg, PA, USA, 2004a. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220355.1220506. URL http://dx.doi.org/10.3115/1220355.1220506.

Pascale Fung and Percy Cheung. Mining Very-Non-Parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM. In EMNLP 04, 2004b.

Pascale Fung and Kenneth Ward Church. K-vec: a new approach for aligning parallel texts. In Proceedings of the 15th conference on Computational linguistics - Volume 2, COLING '94, pages 1096–1102, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/991250.991328. URL http://dx.doi.org/10.3115/991250.991328.

William A. Gale and Kenneth W. Church. Identifying word correspondence in parallel texts. In Proceedings of the workshop on Speech and Natural Language, HLT '91, pages 152–157, Stroudsburg, PA, USA, 1991. Association for Computational Linguistics. doi: 10.3115/112405.112428. URL `http://dx.doi.org/10.3115/112405.112428`.

William A. Gale and Kenneth W. Church. A program for aligning sentences in bilingual corpora. Comput. Linguist., 19:75–102, March 1993. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=972450.972455`.

Ulrich Germann. Aligned hansards of the 36th parliament of canada. Natural Language Group of the USC Information Sciences Institute, 2001a.

Ulrich Germann. Building a statistical machine translation system from scratch: how much bang for the buck can we expect? In Proceedings of the workshop on Data-driven methods in machine translation-Volume 14, pages 1–8. Association for Computational Linguistics, 2001b.

Fei Huang, Ying Zhang, and Stephan Vogel. Mining key phrase translations from web corpora. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pages 483–490, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220636. URL `http://dx.doi.org/10.3115/1220575.1220636`.

Roy Bar-Haim Ido, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo

Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge, 2006.

Martin Kay. The proper place of men and machines in language translation. Machine Translation, 12(1-2):3–23, 1997.

Martin Kay and Martin Röscheisen. Text-translation alignment. Comput. Linguist., 19:121–142, March 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972450.972457.

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

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In hltnaacl, pages 127–133, Edmonton, Canada, May 2003. URL http://people.csail.mit.edu/people/koehn/publications/phrase2003.pdf.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris ohis has led researchers to develop methods for finding Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 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, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1557769.1557821.

J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning, pages 282–289, 2001.

William Nash Locke and Andrew Donald Booth. Machine translation of languages: fourteen essays. Published jointly by Technology Press of the Massachusetts Institute of Technology and Wiley, New York, 1955.

Edward Loper and Steven Bird. Nltk: the natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118117. URL `http://dx.doi.org/10.3115/1118108.1118117`.

Xiaoyi Ma. Chinese english news magazine parallel text. LDC2005T10, 2005.

Robert Moore. Fast and Accurate Sentence Alignment of Bilingual Corpora. In Stephen Richardson, editor, Machine Translation: From Research to Real Users, volume 2499 of Lecture Notes in Computer Science, pages 135–144. Springer Berlin / Heidelberg, 2002. ISBN 978-3-540-44282-0.

Dragos Stefan Munteanu and Daniel Marcu. Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. Comput. Linguist., 31:477–504, December 2005. ISSN 0891-2017. doi: http://dx.doi.org/10.1162/089120105775299168. URL http://dx.doi.org/10.1162/089120105775299168.

Dragos Stefan Munteanu, Alexander Fraser, and Daniel Marcu. Improved Machine Translation Performance via Parallel Sentence Extraction from Comparable Corpora. In NAACL, pages 265–272, 2004.

Jian-Yun Nie, Michel Simard, Pierre Isabelle, and Richard Durand. 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, SIGIR '99, pages 74–81, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1. doi: 10.1145/312624.312656. URL http://doi.acm.org/10.1145/312624.312656.

Franz Josef Och. Minimum error rate training in statistical machine translation. In acl, pages 160–167, Sapporo, Japan, 2003. URL http://acl.ldc.upenn.edu/P/P03/P03-1021.pdf.

Jose Oncina and Marc Sebban. Learning stochastic edit distance: Application in handwritten character recognition. Pattern Recogn., 39:1575–1587, September 2006. ISSN 0031-3203. doi: 10.1016/j.patcog.2006.03.011. URL http://dl.acm.org/citation.cfm?id=1220973.1221331.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In acl, pages 311–318, Philadelpha, Pennsylvania, USA, 2002.

Matt Post, Chris Callison-Burch, and Miles Osborne. Constructing parallel corpora for six indian languages via crowdsourcing. WMT '10, 2012.

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

Philip Resnik. Mining the web for bilingual text. In Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99, pages 527–534, Stroudsburg, PA, USA, 1999. Association for Computational Linguistics. ISBN 1-55860-609-3. doi: 10.3115/1034678.1034757. URL http://dx.doi.org/10.3115/1034678.1034757.

Eric Sven Ristad and Peter N. Yianilos. Learning String-Edit Distance. IEEE Trans. Pattern Anal. Mach. Intell., 20:522–532, May 1998. ISSN 0162-8828. doi: 10.1109/34.682181. URL http://dl.acm.org/citation.cfm?id=279270.279279.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. A dom tree alignment model for mining parallel data from the web. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL-44, pages 489–496, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220237. URL http://dx.doi.org/10.3115/1220175.1220237.

Michel Simard, George F. Foster, and Pierre Isabelle. Using cognates to align sentences in bilingual corpora. In Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing - Volume 2, CASCON '93, pages 1071–1082. IBM Press, 1993. URL http://dl.acm.org/citation.cfm?id=962367.962411.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. Extracting Parallel Sentences from Comparable Corpora using Document Level Alignment. In NAACL 2010, 2010.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In Proceedings of association for machine translation in the Americas, pages 223–231, 2006.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 254–263, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1613715.1613751.

Jörg Tiedemann. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, Recent Advances in Natural Language Processing, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria, 2009. ISBN 978 90 272 4825 1.

C. Tillmann. A Beam-Search extraction algorithm for comparable data. In Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 225–228, 2009.

C. Tillmann and J. Xu. A simple sentence-level extraction algorithm for comparable data. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, pages 93–96, 2009.

Ferhan Ture and Jimmy Lin. Why not grab a free lunch? mining large corpora for parallel sentences to improve translation modeling. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 626–630, Montréal, Canada, June 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N12-1079`.

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. Large scale parallel document mining for machine translation. In Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, pages 1101–

1109, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873905.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz J. Och, and Juri Ganitkevitch. Watermarking the outputs of structured prediction with an application in statistical machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11, pages 1363–1372, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145576.

S. Vogel, H. Ney, and C. Tillmann. HMM-based word alignment in statistical translation. In Proceedings of the 16th conference on Computational linguistics-Volume 2, pages 836–841, 1996.

Wikipedia. Wikipedia, the free encyclopedia, 2004. URL \url{http://en.wikipedia.org/}. [Online; accessed 01-June-2009].

Dekai Wu and Pascale Fung. Inversion Transduction Grammar Constraints for Mining Parallel Sentences from Quasi-Comparable Corpora. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Kwong, editors, Natural Language Processing IJCNLP 2005, volume 3651 of Lecture Notes in Computer Science, pages 257–268. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-29172-5.

Omar F. Zaidan and Chris Callison-Burch. Crowdsourcing translation: professional quality from non-professionals. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 1220–1229, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL http://dl.acm.org/citation.cfm?id=2002472.2002626.

B. Zhao and S. Vogel. Adaptive parallel sentences mining from web bilingual news collection. In Proceedings of the 2002 IEEE International Conference on Data Mining, page 745. IEEE Computer Society, 2002.