Microblogs as Parallel Corpora

Wang Ling[123] **Guang Xiang**[2] **Chris Dyer**[2] **Alan Black**[2] **Isabel Trancoso** [13]
(1)L^{2}F Spoken Systems Lab, INESC-ID, Lisbon, Portugal
(2)Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
(3)Instituto Superior Técnico, Lisbon, Portugal
{lingwang,guangx,cdyer,awb}@cs.cmu.edu
isabel.trancoso@inesc-id.pt

Abstract

In the ever-expanding sea of microblog data, there is a surprising amount of naturally occurring parallel text: some users create post multilingual messages targeting international audiences while others "retweet" translations. We present an efficient method for detecting these messages and extracting parallel segments from them. We have been able to extract over 1M Chinese-English parallel segments from Sina Weibo (the Chinese counterpart of Twitter) using only their public APIs. As a supplement to existing parallel training data, our automatically extracted parallel data yields substantial translation quality improvements in translating microblog text and modest improvements in translating edited news commentary. The resources in described in this paper are available at **www.cs.cmu.edu/%7Elingwang/utopia**.

1 Introduction

Microblogs such as Twitter and Facebook have gained tremendous popularity in the past 10 years. In addition to being an important form of communication for many people, they often contain extremely current, even breaking, information about world events. However, the writing style of microblogs tends to be quite colloquial, with frequent orthographic innovation (*R U still with me or what?*) and nonstandard abbreviations (*idk! shm*)—quite unlike the style found in more traditional, edited genres. This poses considerable problems for traditional NLP tools, which were developed with other domains in mind, which often make strong assumptions about orthographic uniformity (i.e., there is just one way to spell *you*).

Machine translation suffers acutely from the domain-mismatch problem caused by microblog text. On one hand, standard models are probably suboptimal since they (like many models) assume orthographic uniformity in the input. However, more acutely, the data used to develop these systems and train their models is drawn from formal and carefully edited domains, such as parallel web pages and translated legal documents. MT training data seldom looks anything like microblog text.

This paper introduces a method for finding naturally occurring parallel microblog text, which helps address the domain-mismatch problem. Our method is inspired by the perhaps surprising observation that a reasonable number of microblog users tweet "in parallel" in two or more languages. For instance, the American entertainer Snoop Dogg regularly posts parallel messages on Sina Weibo (Mainland China's equivalent of Twitter), for example, *watup Kenny Mayne!! - Kenny Mayne*，最近这么样啊！！, where an English message and its Chinese translation are in the same post, separated by a dash. Our method is able to identify and extract such translations. Briefly, this requires determining if a tweet contains more than one language, if these multilingual utterances contain translated material (or are due to something else, such as code switching), and what the translated spans are.

The paper is organized as follows. Section 2 describes the related work in parallel data extraction. Section 3 presents our model to extract parallel data within the same document. Section 4 describes our extraction pipeline. Section 5 describes the data we gathered from both Sina Weibo (Chinese-English) and Twitter (Chinese-English and Arabic-English). We then present experiments showing that our harvested data not only substantially improves translations of microblog text with existing (and arguably inappropriate) translation models, but that it improves the translation of more traditional MT genres, like newswire. We conclude in Section 6.

2 Related Work

Automatic collection of parallel data is a well-studied problem. Approaches to finding parallel web documents automatically have been particularly important (Resnik and Smith, 2003; Fukushima et al., 2006; Li and Liu, 2008; Uszkoreit et al., 2010; Ture and Lin, 2012). These broadly work by identifying promising candidates using simple features, such as URL similarity or "gist translations" and then identifying truly parallel segments with more expensive classifiers. More specialized resources were developed using manual procedures to leverage special features of very large collections, such as Europarl (Koehn, 2005).

Mining parallel or comparable messages from microblogs has mainly relied on Cross-Lingual Information Retrieval techniques (CLIR). Jelh et al. (2012) attempt to find pairs of tweets in Twitter using Arabic tweets as search queries in a CLIR system. Afterwards, the model described in (Xu et al., 2001) is applied to retrieve a set of ranked translation candidates for each Arabic tweet, which are then used as parallel candidates.

The work on mining parenthetical translations (Lin et al., 2008), which attempts to find translations within the same document, has some similarities with our work, since parenthetical translations are within the same document. However, parenthetical translations are generally used to translate names or terms, which is more limited than our work which extracts whole sentence translations.

Finally, crowd-sourcing techniques to obtain translations have been previously studied and applied to build datasets for casual domains (Zbib et al., 2012; Post et al., 2012). These approaches require remunerated workers to translate the messages, and the amount of messages translated per day is limited. We aim to propose a method that acquires large amounts of parallel data for free. The drawback is that there is a margin of error in the parallel segment identification and alignment. However, our system can be tuned for precision or for recall.

3 Parallel Segment Retrieval

We will first abstract from the domain of Microblogs and focus on the task of retrieving parallel segments from single documents. Prior work on finding parallel data attempts to reason about the probability that pairs of documents $(\mathbf{x}, \mathbf{y})$ are parallel. In contrast, we only consider one document at a time, defined by $\mathbf{x} = x_1, x_2, \ldots, x_n$, and consisting of n tokens, and need to determine whether there is **parallel data** in $\mathbf{x}$, and if so, where are the parallel **segments** and their **languages**. For simplicity, we assume that there are at most 2 continuous segments that are parallel.

As representation for the parallel segments within the document, we use the tuple $([p, q], l, [u, v], r, \mathbf{a})$. The word indexes $[p, q]$ and $[u, v]$ are used to identify the left segment (from p to q) and right segment (from u to v), which are parallel. We shall refer $[p, q]$ and $[u, v]$ as the **spans** of the left and right segments. To avoid overlaps, we set the constraint $p \leq q < u \leq v$. Then, we use l and r to identify the language of the left and right segments, respectively. Finally, $\mathbf{a}$ represents the word alignment between the words in the left and the right segments.

The main problem we address is to find the parallel data when the boundaries of the parallel segments are not defined explicitly. If we knew the indexes $[p, q]$ and $[u, v]$, we could simply run a language detector for these segments to find l and r. Then, we would use an word alignment model (Brown et al., 1993; Vogel et al., 1996), with source $\mathbf{s} = x_p, \ldots, x_q$, target $\mathbf{t} = x_u, \ldots, x_v$ and lexical table $\theta_{l,r}$ to calculate the Viterbi alignment $\mathbf{a}$. Finally, from the probability of the word alignments, we can determine whether the segments are parallel.

Thus, our model will attempt to find the optimal values for the segments $[p, q][u, v]$, languages l, r and word alignments $\mathbf{a}$ jointly. However, there are two problems with this approach. Firstly, word alignment models generally attribute higher probabilities to smaller segments, since these are the result of a smaller product chain of probabilities. In fact, because our model can freely choose the segments to align, choosing only one word as the left segment that is well aligned to a word in the right segment would be the best choice. This is obviously not our goal, since we would not obtain any useful sentence pairs. Secondly, inference must be performed over the combination of all latent variables, which is intractable using a brute force algorithm. We shall describe our model to solve the first problem in 3.1 and our dynamic programming approach to make the inference tractable in 3.2.

3.1 Model

We propose a simple (non-probabilistic) three-factor model that models the spans of the parallel segments, their languages, and word alignments jointly. This model is defined as follows:

$$\begin{aligned} S([u,v],r,[p,q],&l,\mathbf{a} \mid \mathbf{x}) = \\ &S_S^{\alpha}([p,q],[u,v] \mid \mathbf{x})\times \\ &S_L^{\beta}(l,r \mid [p,q],[u,v],\mathbf{x})\times \\ &S_T^{\gamma}(\mathbf{a} \mid [p,q],l,[u,v],r,\mathbf{x}) \end{aligned}$$

Each of the components is weighted by the parameters α, β and γ. We set these values empirically $\alpha = 0.3$, $\beta = 0.3$ and $\gamma = 0.4$, and leave the optimization of these parameters as future work. We discuss the components of this model in turn.

Span score S_S. We define the score of hypothesized pair of spans $[p,q]$, $[u,v]$ as:

$$\begin{aligned} &S_S([p,q],[u,v] \mid \mathbf{x}) = \\ &\frac{(q-p+1)+(v-u+1)}{\sum_{0<p'\leq q'<u'\leq v'\leq n}(q'-p'+1)+(v'-u'+1)}\times \\ &\quad \psi([p,q],[u,v],\mathbf{x}) \end{aligned}$$

The first factor is a distribution over all spans that assigns higher probability to segmentations that cover more words in the document. It is highest for segmentations that cover all the words in the document (this is desirable since there are many sentence pairs that can be extracted but we want to find the largest sentence pair in the document). The function ψ takes on values of 0 or 1 depending on whether certain constraints are violated, these include: parenthetical constraints that enforce that spans must not break text within parenthetical characters and language constraints that ensure that we do break a sequence of Mandarin characters, Arabic words or Latin words.

Language score S_L. The language score $S_L(l,r \mid [p,q],[u,v],\mathbf{x})$ indicates whether the language labels l,r are appropriate to the document contents:

$$\begin{aligned} &S_L(l,r \mid [p,q],[u,v],\mathbf{x}) = \\ &\qquad\qquad \frac{\sum_{i=p}^{q} L(l,x_i) + \sum_{i=u}^{v} L(r,x_i)}{n} \end{aligned}$$

where $L(l,x)$ is a language detection function that yields 1 if the word x_i is in language l, and 0 otherwise. We build the function simply by considering all words that are composed of Latin characters as English, Arabic characters as Arabic and Han characters as Mandarin. This approach is not perfect, but it is simple and works reasonably well for our purposes.

Translation score S_T. The translation score $S_T(\mathbf{a} \mid [p,q],l,[u,v],r)$ indicates whether $[p,q]$ is a reasonable translation of $[u,v]$ with the alignment **a**. We rely on IBM Model 1 probabilities for this score:

$$\begin{aligned} &S_T(\mathbf{a} \mid [p,q],l,[u,v],r,\mathbf{x}) = \\ &\qquad \frac{1}{(q-p+1)^{v-u+2}} \prod_{i=u}^{v} P_{\mathrm{M1}}(x_i \mid x_{a_i}). \end{aligned}$$

The lexical tables P_{M1} for the various language pairs are trained a priori using available parallel corpora. While IBM Model 1 produces worse alignments than other models, in our problem, we need to efficiently consider all possible spans, language pairs and word alignments, which makes the problem intractable. We will show that dynamic programing can be used to make this problem tractable, using Model 1. Furthermore, IBM Model 1 has shown good performance for sentence alignment systems previously (Xu et al., 2005; Braune and Fraser, 2010).

3.2 Inference

Our goal is to find the spans, language pair and alignments such that:

$$\underset{[p,q],l,[u,v],r,\mathbf{a}}{\arg\max}\ S([p,q],l,[u,v],r,\mathbf{a} \mid \mathbf{x}) \qquad (1)$$

A high score indicates that the predicted bispan is likely to correspond to a valid parallel span, so we set a constant threshold τ to determine whether a document has parallel data, i.e., the value of z:

$$z^* = \max_{[u,v],r,[p,q],l,\mathbf{a}} S([u,v],r,[p,q],l,\mathbf{a} \mid \mathbf{x}) > \tau$$

Naively maximizing Eq. 1 would require $O(|\mathbf{x}|^6)$ operations, which is too inefficient to be practical on large datasets. To process millions of documents, this process would need to be optimized.

The main bottleneck of the naive algorithm is finding new Viterbi Model 1 word alignments every time we change the spans. Thus, we propose an iterative approach to compute the Viterbi word alignments for IBM Model 1 using dynamic programming.

Dynamic programming search. The insight we use to improve the runtime is that the Viterbi word alignment of a bispan can be reused to calculate the Viterbi word alignments of larger bispans. The algorithm operates on a 4-dimensional chart of bispans. It starts with the minimal valid span (i.e., $[0, 0], [1, 1]$) and progressively builds larger spans from smaller ones. Let $A_{p,q,u,v}$ represent the Viterbi alignment (under S_T) of the bispan $[p, q], [u, v]$. The algorithm uses the following recursions defined in terms of four operations $\lambda_{\{+v,+u,+p,+q\}}$ that manipulate a single dimension of the bispan to construct larger spans:

- $A_{p,q,u,v+1} = \lambda_{+v}(A_{p,q,u,v})$ adds one token to the end of the right span with index $v + 1$ and find the viterbi alignment for that token. This requires iterating over all the tokens in the left span, $[p, q]$ and possibly updating their alignments. See Fig. 1 for an illustration.

- $A_{p,q,u+1,v} = \lambda_{+u}(A_{p,q,u,v})$ removes the first token of the right span with index u, so we only need to remove the alignment from u, which can be done in time $O(1)$.

- $A_{p,q+1,u,v} = \lambda_{+q}(A_{p,q,u,v})$ adds one token to the end of the left span with index $q + 1$, we need to check for each word in the right span, if aligning to the word in index $q + 1$ yields a better translation probability. This update requires $n - q + 1$ operations.

- $A_{p+1,q,u,v} = \lambda_{+p}(A_{p,q,u,v})$ removes the first token of the left span with index p. After removing the token, we need to find new alignments for all tokens that were aligned to p. Thus, the number of operations for this update is $K \times (q - p + 1)$, where K is the number of words that were aligned to p. In the best case, no words are aligned to the token in p, and we can simply remove it. In the worst case, if all target words were aligned to p, this update will result in the recalculation of all Viterbi Alignments.

The algorithm proceeds until all valid cells have been computed. One important aspect is that the update functions differ in complexity, so the sequence of updates we apply will impact the performance of the system. Most spans are reachable using any of the four update functions. For instance, the span $A_{2,3,4,5}$ can be reached using $\lambda_{+v}(A_{2,3,4,4})$, $\lambda_{+u}(A_{2,3,3,5})$, $\lambda_{+q}(A_{2,2,4,5})$ or $\lambda_{+p}(A_{1,3,4,5})$. However, we want to use λ_{+u} whenever possible, since it only requires one operation, although that is not always possible. For instance, the state $A_{2,2,2,4}$ cannot be reached using λ_{+u}, since the state $A_{2,2,1,4}$ is not valid, because the spans overlap. If this happens, incrementally more expensive updates need to be used, such as λ_{+v}, then λ_{+q}, which are in the same order of complexity. Finally, we want to minimize the use of λ_{+p}, which is quadratic in the worst case. Thus, we use the following recursive formulation that guarantees the optimal outcome:

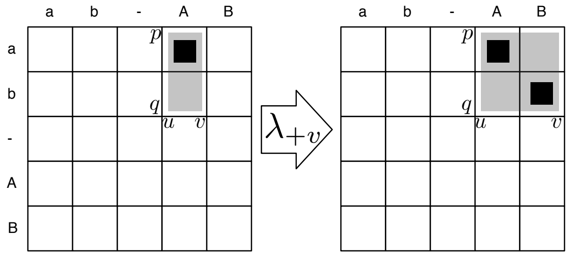

Figure 1: Illustration of the λ_{+v} operator. The light gray boxes show the parallel span and the dark boxes show the span's Viterbi alignment. In this example, the parallel message contains a "translation" of a b to A B.

$$A_{p,q,u,v} = \begin{cases} \lambda_{+u}(A_{p,q,u-1,v}) & \text{if } u > q + 1 \\ \lambda_{+v}(A_{p,q,u,v-1}) & \text{else if } v > q + 1 \\ \lambda_{+p}(A_{p-1,q,u,v}) & \text{else if } q = p + 1 \\ \lambda_{+q}(A_{p,q-1,u,v}) & \text{otherwise} \end{cases}$$

This transition function applies the cheapest possible update to reach state $A_{p,q,u,v}$.

Complexity analysis. We can see that λ_{+u} is only needed in the following the cases $[0, 1][2, 2], [1, 2][3, 3], \cdots, [n - 2, n - 1][n, n]$. Since, this update is quadratic in the worst case, the complexity of this operations is $O(n^3)$. The update λ_{+q}, is applied to the cases $[*, 1][2, 2], [*, 2][3, 3], \cdots, [*, n-1], [n, n]$, where $*$ denotes any number within the span constraints but not present in previous updates. Since, the update is linear and we need to iterate through all tokens twice, this update takes $O(n^3)$ operations. The update λ_{+v} is applied for the cases $[*, 1][2, *], [*, 2][3, *], \cdots, [*, n - 1], [n, *]$. Thus, with three degrees of freedom and a linear update, it runs in $O(n^4)$ time. Finally, update λ_{+u} runs in constant time, but is run for all remaining cases, which constitute $O(n^4)$ space. By summing the executions of all updates, we observe that the order of magnitude of our exact inference process is $O(n^4)$. Note that for exact inference, it is not possible to get a lower order of magnitude, since we need to at least iterate through all possible span values once, which takes $O(n^4)$ time.

4 Parallel Data Extraction

We will now describe our method to extract parallel data from Microblogs. The target domains in this work are Twitter and Sina Weibo, and the main language pair is Chinese-English. Furthermore, we also run the system for the Arabic-English language pair using the Twitter data.

For the Twitter domain, we use a previously crawled dataset from the years 2008 to 2013, where one million tweets are crawled every day. In total, we processed 1.6 billion tweets.

Regarding Sina Weibo, we built a crawler that continuously collects tweets from Weibo. We start from one seed user and collect his posts, and then we find the users he follows that we have not considered, and repeat. Due to the rate limiting established by the Weibo API[1], we are restricted in terms of number of requests every hour, which greatly limits the amount of messages we can collect. Furthermore, each request can only fetch up to 100 posts from a user, and subsequent pages of 100 posts require additional API calls. Thus, to optimize the number of parallel posts we can collect per request, we only crawl all messages from users that have at least 10 parallel tweets in their first 100 posts. The number of parallel messages is estimated by running our alignment model, and checking if $\tau > \phi$, where ϕ was set empirically initially, and optimized after obtaining annotated data, which will be detailed in 5.1. Using this process, we crawled 65 million tweets from Sina Weibo within 4 months.

In both cases, we first filter the collection of tweets for messages containing at least one trigram in each language of the target language pair, determined by their Unicode ranges. This means that for the Chinese-English language pair, we only keep tweets with more than 3 Mandarin characters and 3 latin words. Furthermore, based on the work in (Jelh et al., 2012), if a tweet A is identified as a retweet, meaning that it references another tweet B, we also consider the hypothesis that these tweets may be mutual translations. Thus, if A and B contain trigrams in different languages, these are also considered for the extraction of parallel data. This is done by concatenating tweets A and B, and adding the constraint that $[p, q]$ must be within A and $[u, v]$ must be within B. Finally, identical duplicate tweets are removed.

After filtering, we obtained 1124k ZH-EN tweets from Sina Weibo, 868k ZH-EN and 136k AR-EN tweets from Twitter. These language pairs are not definite, since we simply check if there is a trigram in each language.

Finally, we run our alignment model described in section 3, and obtain the parallel segments and their scores, which measure how likely those segments are parallel. In this process, lexical tables for EN-ZH language pair used by Model 1 were built using the FBIS dataset (LDC2003E14) for both directions, a corpus of 300K sentence pairs from the news domain. Likewise, for the EN-AR language pair, we use a fraction of the NIST dataset, by removing the data originated from UN, which leads to approximately 1M sentence pairs.

5 Experiments

We evaluate our method in two ways. First, intrinsically, by observing how well our method identifies tweets containing parallel data, the language pair and what their spans are. Second, extrinsically, by looking at how well the data improves a translation task. This methodology is similar to that of Smith et al. (2010).

5.1 Parallel Data Extraction

Data. Our method needs to determine if a given tweet contains parallel data, and if so, what is the language pair of the data, and what segments are parallel. Thus, we had a native Mandarin speaker, also fluent in English, to annotate 2000 tweets sampled from crawled Weibo tweets. One important question of answer is what portion of the Microblogs contains parallel data. Thus, we also use the random sample Twitter and annotated 1200 samples, identifying whether each sample contains parallel data, for the EN-ZH and AR-EN filtered tweets.

Metrics. To test the accuracy of the score S, we ordered all 2000 samples by score. Then, we calculate the precision, recall and accuracy at increasing intervals of 10% of the top samples. We count

[1] http://open.weibo.com/wiki/API文档/en

as a true positive (tp) if we correctly identify a parallel tweet, and as a false positive (fp) spuriously detect a parallel tweet. Finally, a true negative (tn) occurs when we correctly detect a non-parallel tweet, and a false negative (fn) if we miss a parallel tweet. Then, we set the precision as $\frac{tp}{tp+fp}$, recall as $\frac{tp}{tp+fn}$ and accuracy as $\frac{tp+tn}{tp+fp+tn+fn}$. For language identification, we calculate the accuracy based on the number of instances that were identified with the correct language pair. Finally, to evaluate the segment alignment, we use the Word Error Rate (WER) metric, without substitutions, where we compare the left and right spans of our system and the respective spans of the reference. We count an insertion error (I) for each word in our system's spans that is not present in the reference span and a deletion error (D) for each word in the reference span that is not present in our system's spans. Thus, we set $WER = \frac{D+I}{N}$, where N is the number of tokens in the tweet. To compute this score for the whole test set, we compute the average of the WER for each sample.

Results. The precision, recall and accuracy curves are shown in Figure 2. The quality of the parallel sentence detection did not vary significantly with different setups, so we will only show the results for the best setup, which is the baseline model with span constraints.

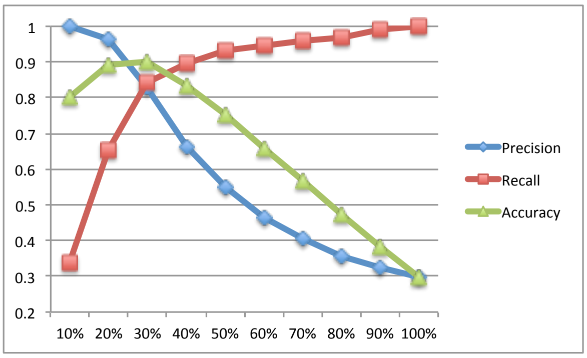

Figure 2: Precision, recall and accuracy curves for parallel data detection. The y-axis denotes the scores for each metric, and the x-axis denotes the percentage of the highest scoring sentence pairs that are kept.

From the precision and recall curves, we observe that most of the parallel data can be found at the top 30% of the filtered tweets, where 5 in 6 tweets are detected correctly as parallel, and only 1 in every 6 parallel sentences is lost. We will denote the score threshold at this point as ϕ, which is a good threshold to estimate on whether the tweet is parallel. However, this parameter can be tuned for precision or recall. We also see that in total, 30% of the filtered tweets are parallel. If we generalize this ratio for the complete set with 1124k tweets, we can expect approximately 337k parallel sentences. Finally, since 65 million tweets were extracted to generate the 337k tweets, we estimate that approximately 1 parallel tweet can be found for every 200 tweets we process using our targeted approach. On the other hand, from the 1200 tweets from Twitter, we found that 27 had parallel data in the ZH-EN pair, if we extrapolate for the whole 868k filtered tweets, we expect that we can find 19530. 19530 parallel sentences from 1.6 billion tweets crawled randomly, represents 0.001% of the total corpora. For AR-EN, a similar result was obtained where we expect 12407 tweets out of the 1.6 billion to be parallel. This shows that targeted approaches can substantially reduce the crawling effort required to find parallel tweets. Still, considering that billions of tweets are posted daily, this is a substantial source of parallel data. The remainder of the tests will be performed on the Weibo dataset, which contains more parallel data. Tests on the Twitter data will be conducted as future work, when we process Twitter data on a larger scale to obtain more parallel sentences.

For the language identification task, we had an accuracy of 99.9%, since distinguishing English and Mandarin is trivial. The small percentage of errors originated from other latin languages (Ex: French) due to our naive language detector.

As for the segment alignment task. Our baseline system with no constraints obtains a WER of 12.86%, and this can be improved to 11.66% by adding constraints to possible spans. This shows that, on average, approximately 1 in 9 words on the parallel segments is incorrect. However, translation models are generally robust to such kinds of errors and can learn good translations even in the presence of imperfect sentence pairs.

Among the 578 tweets that are parallel, 496 were extracted within the same tweet and 82 were extracted from retweets. Thus, we see that the majority of the parallel data comes from within the same tweet.

Topic analysis. To give an intuition about the contents of the parallel data we found, we looked at the distribution over topics of the parallel dataset inferred by LDA (Blei et al., 2003). Thus,

Topic	Most probable words in topic
1 (Dating)	love time girl live mv back word night rt wanna
2 (Entertainment)	news video follow pong image text great day today fans
3 (Music)	cr day tour cn url amazon music full concert alive
4 (Religion)	man god good love life heart would give make lord
5 (Nightlife)	cn url beijing shanqi party adj club dj beijiner vt
6 (Chinese News)	china chinese year people world beijing years passion country government
7 (Fashion)	street fashion fall style photo men model vogue spring magazine

Table 1: Most probable words inferred using LDA in several topics from the parallel data extracted from Weibo. Topic labels (in parentheses) were created manually for illustration purposes.

we grouped the Weibo filtered tweets by users, and ran LDA over the predicted English segments, with 12 topics. The 7 most interpretable topics are shown in Table 1. We see that the data contains a variety of topics, both formal (Chinese news, religion) and informal (entertainment, music).

5.2 Machine Translation Experiments

We report on machine translation experiments using our harvested data in two domains: edited news and microblogs.

News translation. For the news test, we created a new test set from a crawl of the Chinese-English documents on the Project Syndicate website[2], which contains news commentary articles. We chose to use this data set, rather than more standard NIST test sets to ensure that we had recent documents in the test set (the most recent NIST test sets contain documents published in 2007, well before our microblog data was created). We extracted 1386 parallel sentences for tuning and another 1386 sentences for testing, from the manually aligned segments. For this test set, we used 8 million sentences from the full NIST parallel dataset as the language model training data. We shall call this test set **Syndicate**.

Microblog translation. To carry out the microblog translation experiments, we need a high quality parallel test set. Since we are not aware of such a test set, we created one by manually selecting parallel messages from Weibo. Our procedure was as follows. We selected 2000 candidate Weibo posts from users who have a high number of parallel tweets according to our automatic method (at least 2 in every 5 tweets). To these, we added another 2000 messages from our targeted Weibo crawl, but these had no requirement on the proportion of parallel tweets they had produced. We identified 2374 parallel segments, of which we used 1187 for development and 1187 for testing. We refer to this test set as **Weibo**.[3]

Obviously, we removed the development and test sets from our training data. Furthermore, to ensure that our training data was not too similar to the test set in the Weibo translation task, we filtered the *training* data to remove near duplicates by computing edit distance between each parallel sentence in the heldout set and each training instance. If either the source or the target sides of the a training instance had an edit distance of less than 10%, we removed it.[4] As for the language models, we collected a further 10M tweets from Twitter for the English language model and another 10M tweets from Weibo for the Chinese language model.

Baselines. We report results on these test sets using different training data. First, we use the FBIS dataset which contains 300K high quality sentence pairs, mostly in the broadcast news domain. Second, we use the full 2012 NIST Chinese-English dataset (approximately 8M sentence pairs, including FBIS). Finally, we use our crawled data (referred as Weibo) by itself and also combined with the two previous training sets.

Setup. We use the Moses phrase-based MT system with standard features (Koehn et al., 2003). For reordering, we use the MSD reordering model (Axelrod et al., 2005). As the language model, we use a 5-gram model with Kneser-Ney smoothing. The weights were tuned using MERT (Och, 2003). Results are presented with BLEU-4 (Papineni et al., 2002).

[2] http://www.project-syndicate.org/

[3] We acknowledge that self-translated messages are probably not a typically representative sample of all microblog messages. However, we do not have the resources to produce a carefully curated test set with a more broadly representative distribution. Still, we believe these results are informative as long as this is kept in mind.

[4] Approximately 150,000 training instances removed.

Syndicate (test)			**Weibo** (test)		
FBIS	NIST	Weibo	FBIS	NIST	Weibo
obama (83)	barack (59)	democracies (15)	2012 (24)	showstudio (9)	submissions (4)
barack (59)	namo (6)	imbalances (13)	alanis (13)	crue (9)	ivillage (4)
princeton (40)	mitt (6)	mahmoud (12)	crue (9)	overexposed (8)	scola (3)
ecb (8)	guant (6)	millennium (9)	showstudio (9)	tweetmeian (5)	rbst (3)
bernanke (8)	fairtrade (6)	regimes (8)	overexposed (8)	tvd (5)	curitiba (3)
romney (7)	hollande (5)	wolfowitz (7)	itunes (8)	iheartradio (5)	zeman (2)
gaddafi (7)	wikileaks (4)	revolutions (7)	havoc (8)	xoxo (4)	@yaptv (2)
merkel (7)	wilders (3)	qaddafi (7)	sammy (6)	snoop (4)	witnessing (2)
fats (7)	rant (3)	geopolitical (7)	obama (6)	shinoda (4)	whoohooo (2)
dialogue (7)	esm (3)	genome (7)	lol (6)	scrapbook (4)	wbr (2)

Table 3: The most frequent out-of-vocabulary (OOV) words and their counts for the two English-source test sets with three different training sets.

	Syndicate		**Weibo**	
	ZH-EN	EN-ZH	ZH-EN	EN-ZH
FBIS	9.4	18.6	10.4	12.3
NIST	11.5	21.2	11.4	13.9
Weibo	8.75	15.9	15.7	17.2
FBIS+Weibo	11.7	19.2	16.5	17.8
NIST+Weibo	**13.3**	**21.5**	**16.9**	**17.9**

Table 2: BLEU scores for different datasets in different translation directions (left to right), broken with different training corpora (top to bottom).

Results. The BLEU scores for the different parallel corpora are shown in Table 2 and the top 10 out-of-vocabulary (OOV) words for each dataset are shown in Table 3. We observe that for the **Syndicate** test set, the NIST and FBIS datasets perform better than our extracted parallel data. This is to be expected, since our dataset was extracted from an extremely different domain. However, by combining the Weibo parallel data with this standard data, improvements in BLEU are obtained. Error analysis indicates that one major factor is that names from current events, such as *Romney* and *Wikileaks* do not occur in the older NIST and FBIS datasets, but they are represented in the Weibo dataset. Furthermore, we also note that the system built on the Weibo dataset does not perform substantially worse than the one trained on the FBIS dataset, a further indication that harvesting parallel microblog data yields a diverse collection of translated material.

For the **Weibo** test set, a significant improvement over the news datasets can be achieved using our crawled parallel data. Once again newer terms, such as *iTunes*, are one of the reasons older datasets perform less well. However, in this case, the top OOV words of the news domain datasets are not the most accurate representation of coverage problems in this domain. This is because many frequent words in microblogs, such as abbreviations, like *u* and *4* are found in the news domain as source words, but with different meanings. Thus, the OOV table gives an incomplete picture of the translation problems words when using the news domain corpora to translate microblogs. Finally, we observe that combining the datasets yields another gain over individual datasets.

6 Conclusions

We presented a framework to crawl parallel data from microblogs. The model finds parallel data from single documents, with translations of the same sentence in two languages. Using this approach, we show that a considerable amount of parallel sentence pairs can be crawled from microblogs and these can be used to improve Machine Translation by updating our translation tables with translations of newer terms. Furthermore, the in-domain data can substantially improve the translation quality on microblog data.

As future work, we shall scale our method to larger microblog datasets and more language pairs, with emphasis on low resource languages.

The resources described in this paper and further developments are available to the general public at **www.cs.cmu.edu/%7Elingwang/utopia**.

Acknowledgements

The PhD thesis of Wang Ling is supported by FCT grant SFRH/BD/51157/2010. The authors also wish to thank William Cohen, Noah Smith and the anonymous reviewers for their insight and comments. We are also extremely grateful to Brendan O'Connor for providing the Twitter data.

References

Amittai Axelrod, Ra Birch Mayne, Chris Callison-burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Fabienne Braune and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 81–89, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, June.

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

Laura Jelh, Felix Hiebel, and Stefan Riezler. 2012. Twitter translation using translation-based cross-lingual retrieval. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 410–421, Montréal, Canada, June. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

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

Bo Li and Juan Liu. 2008. Mining Chinese-English parallel corpora from the web. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*.

Dekang Lin, Shaojun Zhao, Benjamin Van Durme, and Marius Paşca. 2008. Mining parenthetical translations from the web by word alignment. In *Proceedings of ACL-08: HLT*, pages 994–1002, Columbus, Ohio, June. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409, Montréal, Canada, June. Association for Computational Linguistics.

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

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Ferhan Ture and Jimmy Lin. 2012. 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. Association for Computational Linguistics.

Jakob Uszkoreit, Jay Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1101–1109.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics - Volume 2*, COLING '96, pages 836–841, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jinxi Xu, Ralph Weischedel, and Chanh Nguyen. 2001. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 105–110, New York, NY, USA. ACM.

Jia Xu, Richard Zens, and Hermann Ney. 2005. Sentence segmentation using ibm word alignment model 1. In *Proceedings of EAMT 2005 (10th Annual Conference of the European Association for Machine Translation*, pages 280–287.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwarz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine translation of Arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.