Bilingual Lexicon Induction for Low-resource Languages

July 26, 2010

Abstract

Statistical machine translation relies on the availability of substantial amounts of human translated texts. Such bilingual resources are available for relatively few language pairs, which presents obstacles to applying current statistical translation models to low-resource languages. In this work, we induce bilingual dictionaries from more plentiful monolingual corpora using a diverse set of cues, including: cross-lingual vector space models, the frequencies of words over time, orthographic similarity, etc. We report the efficacy of these monolingual cues and contrast their performance for a language pair where plentiful bilingual resources are available. We further evaluate the accuracy of bilingual dictionaries induced between English and a set of low resource languages. Since our principal objective is to induce wide coverage lexicons, we contrast the performance of our framework on randomly selected source words with an optimistic results obtained on frequent words and typically reported in lexicon induction literature. Finally, we propose a simple and effective technique for using crowd sourced annotations to incrementally refine the output of our lexicon induction system.

1 Introduction

Statistical methods for machine translation continue to push the state of the art in automatic translation. However, they crucially rely on the availability of large numbers of translations aligned across two languages. Generation of these parallel corpora require the efforts of bilingual speakers and are extremely expensive to produce in sufficient quantities to induce a high quality statistical translation system. As a result, these methods can not be successfully applied to the majority of word's languages and especially to low-resource or less frequently taught languages. The DARPA BABEL (Bayesian Architecture Begetting Every Language) project aims to address the problem of scarcity of bilingual training data. In this report, we summarize the progress made in eight and a half months since the start of the project, and outline the next steps we are planning to take.

The first objective of the BABEL project is to (a) collect large monolingual datasets for a number of low resource languages and (b) utilize them to build alternative bilingual resources for inducing translation models in the absence of large parallel corpora. In particular, we posit that cheap to collect monolingual resources contain cues which can be utilized for inducing bilingual lexicons. In this report, we define and exploit a handful of these cues to generate wide coverage dictionaries between English and a large set of low density languages. Prior work on lexicon induction has attempted to automatically learn bilingual lexicons, either by using monolingual corpora ([Rapp, 1999], [Koehn and Knight, 2002], [Schafer and Yarowsky, 2002], [Haghighi et al., 2008]) or by exploiting the cross-language evidence of closely related bridge languages that have more resources [Mann and Yarowsky, 2001]. While this work falls into the former category, our objective is substantially different. Unlike the bulk of the previous work, our specific aim is to use monolingual cues to induce *broad coverage* dictionaries for a large set of low resource languages. In sum, in this report we summarize the progress toward the first objective of the project, i.e.:

- Describe monolingual resources we are continuously collecting.
- Define a set of cues which we use to induce bilingual lexicons from those resources.
- Propose an evaluation method and show preliminary results on a high resource language pair and ten low resource languages and English.
- Propose an effective technique for using crowd sourced annotations to incrementally refine the output of our lexicon induction system.
- Briefly describe implementation details of the induction framework.

The second goal of the BABEL project is to propose novel translation models to utilize these monolingual resources in the absence of large explicitly aligned bilingual corpora. In this report, we also describe the ongoing work toward introducing these models.

2 Related Work

[Rapp, 1995] was the first to propose using context of a given word as a clue to its translation. Given a German word with an unknown translation, its surrounding words were collected and translated into English using a small seed dictionary. Words with similar context in a monolingual English corpus were then proposed as translation candidates. The original work employed a bilingual dictionary containing approximately 16,000 words. Contextual vectors were collected from corpora of 135 and 163 million words on the German and English sides, respectively. While our goal is collect and exploit large monolingual corpora, we do not yet have access to large datasets for all of low resource languages we consider (see Section 4). Moreover, [Rapp, 1995] lemmatized both monolingual corpora, a step we cannot generally assume in our low-resource setting (we discuss alternatives in Section 3.3). Exact test dictionary matching on a small collection of 100 manually selected nouns was used to evaluate the induction procedure. We show (Section 5.1) that evaluation results on a small set of hand selected words reported in this in most of the previous work can be misleading if the objective is to induce large translation dictionaries. Instead, we focus on measuring the quality of induced lexicons on *randomly* selected words.

Subsequent work has explored Rapp's ideas proposing a variety of alternative similarity metrics, better methods for collecting context, and novel monolingual cues. It can be roughly grouped as follows:

- [Rapp, 1999, Fung and Yee, 1998] directly extend the original idea proposing alternative similarity metrics. We will follow this line of work when defining the contextual cue in Section 3.
- [Garera et al., 2009] show further improvements when contexts are derived from dependency trees rather than using adjacency. We plan to include dependency context information whenever available into our current implementation.
- [Schafer and Yarowsky, 2002] exploited the idea that word translations tend to co-occur in time across languages. [Klementiev and Roth, 2006] exploited the temporal cue to train a phonetic similarity model for associating Named Entities across languages and used a Discrete Fourier Transform similarity metric novel in this domain. We also use this cue for associating words in time stamped newswire data in Section 3.
- [Mimno et al., 2009] propose a polylingual topic model and match high probability words in each topic across languages. While effective for a small number of topics (448 in their experiments), it is unlikely to scale well for inducing large multilingual dictionaries. We propose an alternative to exploit explicit category information already present in many monolingual resources (such as news stories and Wikipedia articles) in Section 3.

Keep topic?

- A few authors have exploited strategies for combining multiple cues typically employing simple heuristics (e.g. [Schafer and Yarowsky, 2002] and [Koehn and Knight, 2002]). We use a simple information retrieval scheme for combining ranked lists of translation candidates, and work to exploit more sophisticated methods (e.g. [Klementiev and Roth, 2006] and [Klementiev et al., 2008]) is ongoing.
- Not directly related to lexicon induction is the work on exploiting crowd sourced annotations in various NLP tasks (e.g. [Snow et al., 2008], and [Callison-Burch, 2009]). We propose an effective technique for using crowd sourced annotations to incrementally refine the output of our lexicon induction system

A few words about the scalability of Haghighli's model.

3 Inducing Bilingual Lexicons From Monolingual Cues

Various linguistic and corpus cues can be informative for relating word translations across a pair of languages. We begin by considering a few of them and explain how they can be used to measure similarity between a source word and a translation candidate.

Little technical details, i.e. no metric defs, etc. Add?

3.1 Monolingual cues

Contextual cue. [Rapp, 1999] proposed inducing a translation dictionary from disparate monolingual texts. They populate a bilingual lexicon by projecting *context vectors* across two languages using vector-space semantic models to represent words [Deerwester et al., 1990]. The elements in vector-based representations of a word indicate the frequency of its co-occurrence with all other words in the same language. For instance, the vector representation of "airplane" would indicate that it frequently occurs in contexts near the words "airport", "flight", "landing," "passengers", "pilot", "runway", etc. The similarities of words within one language can be measured using the distance between their vectors, with cosine similarity, for instance. To translate unknown words, [Rapp, 1999] suggests building vector space models of two languages.

More formally, assume that $(s_1, s_2, \dots s_N)$ and $(t_1, t_2, \dots t_M)$ are (arbitrarily indexed) source and target vocabularies, respectively. Each source word s_i (target word t_j) is represented with an N (M) dimensional vector. Only the components corresponding to words that appear in the context of s_i (t_j) in data take on non-zero values, which typically measure how "unique" a word is to the context in the dataset. Next, the source vectors are projected by mapping each source component to a component in the target space corresponding to its dictionary translation, but retaining the source component value. This sparse projected vector is compared to the vectors for all words in the target language, and a similarity score is used to select the best candidate or induce a ranking. Various means of computing the component values and vector similarity measures have been proposed in literature (e.g. [Rapp, 1999, Fung and Yee, 1998]). While the quality of the resulting induced lexicon depends on the data, we found the following to work best in our experiments. We compute the value of the k-th component of s_i's contextual vector as follows:

$$w_k^{(i)} = n_{i,k} \times (log(n/n_i) + 1) \tag{1}$$

where $n_{i,k}$ and s_k are the number of times s_k appears in the context of s_i and in the entire corpus, and n is the maximum number of occurrences of any word in the data. Intuitively, the more frequently s_k appears with s_i and the less common it is in the corpus in general, the higher its component value. The similarity between two resulting vectors is measured as a cosine of the angle

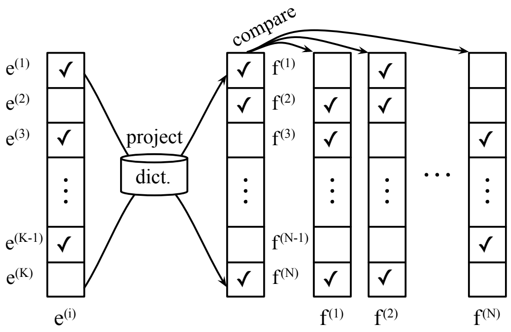

Figure 1: Lexicon induction using contextual information. First, contextual vectors are projected using a small seed dictionary and then compared with the target language candidates.

between them. The entire process is illustrated on Figure 1.

Temporal cue. Online content is often published along with temporal information: news feeds, for example, are comprised of news stories annotated with date and time of publication. The feeds are specialized for the target geographical locations and vary in content across languages. Still, many events are deemed relevant to multiple audiences and the news stories related to them appear in several languages, although rarely as direct translations of one another. Words associated with these events will appear with increased frequency in multiple languages around the dates when these events are reported. Figure 2 illustrates this idea with temporal histograms of three English words and their Spanish translations. Such weak synchronicity provides a cue about the relatedness of words across the two languages, and can be exploited to associate them. In order to score a pair of entities across languages, we can compute the similarity of their temporal signatures. To generate a time sequence for a given word, we first sort the set of (time-stamped) documents of our corpus into a sequence of equally sized temporal bins. We then count the number of occurrences of the word in each bin. Changing the size of the bin or computing counts in a sliding window instead can recover some accuracy if the temporal alignment between two languages in our dataset is poor [Klementiev and Roth, 2006]. Finally, we normalize the sequence and use either the cosine measure or F-index [Hetland, 2004] to score similarity. To compute the latter, we first run a Discrete Fourier Transform on a time sequence to extract its Fourier expansion coefficients. The score of a pair of time sequences is then computed as a Euclidean distance between their expansion coefficient vectors.

Figure 2 is from Europarl, add the one from the proposal instead.

Orthographic cue. Etymologically related words often retain similar spelling across languages with the same writing system. Capturing these similarities

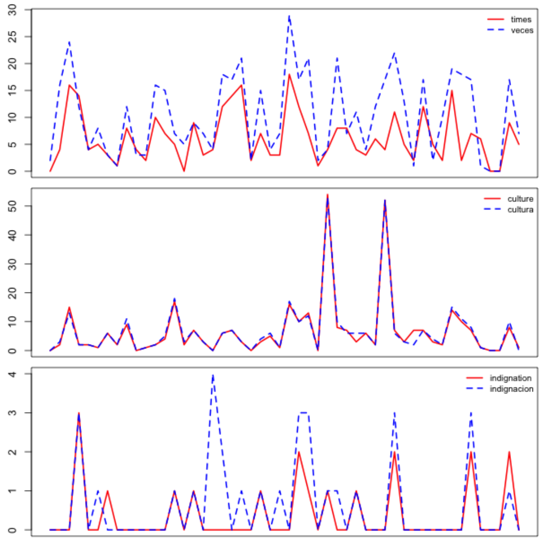

Figure 2: Temporal histograms of three English words and their Spanish translations.

can provide yet another clue about their relatedness. Edit distance defines one such metric, counting the minimal number of edit operations required to transform one string into another. While it won't provide a good signal for most translation pairs, it proves to be a highly effective for related languages such as English and Spanish (see Section 5.1).

Phonetic cue. We can extend the previous idea to language pairs not sharing the same writing system, since many cognates and transliterated words are phonetically similar. [Klementiev and Roth, 2006] train a transliteration model to return high scores to phonetically similar word pairs and show it to be highly successful for Named Entities which are often transliterated. Adding this cue should provide an informative signal for cognates as well.

3.2 Combination strategies

These cues provide informative and independent means to score a source word and a target candidate, and combining them is likely to produce better lexicons. One idea is to directly combine the ranked lists of induced candidates using the mean reciprocal rank (MRR) heuristic frequently used in Information Retrieval literature. The idea is to score each of the translation candidates with an average reciprocal rank across all rankings, and then sort the candidates in descending order.

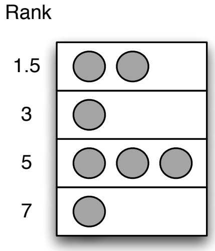

Figure 3: Resolving ties.

The relative informativeness of the cues will depend on the data: e.g. the temporal cue depends on the temporal alignment of of both sides of the bilingual corpus, the orthographic cue is uninformative when the two languages use different scripts, etc. However, the MRR heuristic assigns equal weights to each ranking and investigating more suitable combination strategies remains the subject of ongoing work.

Some of the metrics we have introduced (e.g. edit distance) are likely to assign same scores to multiple candidates. We use the following strategy for resolving ties: (1) a set of candidates assigned the same score shares the same rank, (2) a rank assigned to a candidate (a real value) takes into account the number of other candidates with the same or better scores (Figure 3).

3.3 Handling morphology

Many of the languages in our list such as Russian and Korean are characterized by morphological rules generating a large number of word forms for the same lexeme. Thus, we need to be able to group morphological variants of the same word into an equivalence class and collect their aggregate statistics. For instance, we would like to count the total number of occurrences of {*Herzegovina, Hercegovina*} on the English side in order to map it accurately to the corresponding equivalence class we may see on the Russian side of our corpus (e.g. {*Герцеговина, Герцеговину, Герцеговины, Герцеговиной*}). In order to keep to our objective of requiring as little language knowledge as possible, we take a rather simplistic approach for grouping equivalence classes. For morphologically rich languages, a set of words are in the same equivalence class if they share a prefix of size five or longer. For the other languages, each unique word is assigned its own class. More sophisticated unsupervised approaches (i.e. [Snyder and Barzilay, 2008, Poon et al., 2009]) could be incorporated instead, and are a subject of ongoing work.

4 Resources

In this section we briefly describe each of the resources we use for inducing translation lexicons. For those being continuously collected, we give a current snapshot of relevant statistics.

Wikipedia. Wikipedia provides a large repository of continuously edited monolingual texts in a large number of languages. Besides the article content, it also contains various metadata which can be useful for generating additional resources. For example, interlingual links can be used to select articles discussing the same subject across multiple languages. We will use these comparable articles when inducing translations using the contextual cue in Section 5. Table 1 lists counts of such page pairs between English and 42 other languages along with the second language scripts used predominantly[1] in the corresponding wikis.

Bilingual Dictionaries. Dictionaries provide a crucial resource for projecting vector-base representations of equivalence classes and for evaluating the quality of induced lexicons. Table 1 lists the sizes of our dictionaries along with their coverage of the English side of the "comparable" bilingual Wikipedia subsets in column 2. Type coverage measures the proportion of unique Wikipedia English words which have an entry in the dictionary, while token coverage takes into account their corpus counts. Low type coverage for most languages is primarily due to misspelled or incorrectly extracted article text in Wikipedia articles generating a long tail of low frequency words, and noise in the dictionaries themselves. On the target side, dictionary noise and rich morphology of some of the languages we consider also substantially reduces coverage. Moreover, some of the dictionaries contain romanized translations (boldfaced in the last column of Table 1) and are not used in the results we present here. Investigating automatic techniques to induce the original language script from dictionary romanization is a subject of future work. We propose an alternative method of obtaining small translation lexicons in Section 5.4.

Stopword lists. We have also collected stop words for Farsi, Bangla, Hindi, Polish, Spanish, Russian, Romanian, and English.

News wire. We have assembled additional monolingual corpora by continuously crawling a set of regularly updated news websites. Up to 10 years of collected and processed data are listed in Table 2 for 25 languages: the last page counts a number of tokens collected and the previous columns list the number of days with at least one associated story for each of the 10 years. We used a set of heuristics to extract language and temporal information from page URL and body text; a substantial portion of the total of 1.4 billion tokens of crawled data not included in the table still remains to be processed for language and time.

Note: Ben's runs used an older crawl from April.

[1] While Uzbek, Serbian, Azeri, Kurdish, Kazakh, Uighur, and Somali use multiple writing systems, we indicate only those used for the bulk of Wikipedia articles.

	Wikipedia			Dictionary			
Language	page	trg tokens	script	all	single	src type/tok	script
	pairs	(× 1000)		entries	words	cover. (%)	
Tigrinya	36	3	o	56	45	0.3/0.6	**r**
Punjabi	401	101	o	76,311	57,539	12.2/57.7	o/**r**
Kyrgyz	492	77	c	74,890	22,510	2.6/42.4	**r**
Somali	585	82	r	230	182	0.2/6.8	r
Nepali	1,293	262	d	6,812	5,499	3.6/39.7	**r**
Tibetan	1,358	35	i	59,083	8,068	2.5/35.1	**r**
Uighur	1,814	114	a/r	16,285	13,444	1.7/37.9	r
Maltese	1,896	706	r	7,574	4,814	2.9/44.7	r
Turkmen	3,137	104	r	91,928	45,430	4.6/54	r
Kazakh	3,470	606	c	145,750	38,066	1/34	**r**
Mongolian	4,009	847	c	948	714	0.3/20.7	**r**
Tatar	4,180	313	c/r	8,557	6,633	1.7/33	c/r
Kurdish	5,059	872	r	9,870	8,416	1.2/32	r
Uzbek	5,875	747	r/c	190,688	94,747	6.2/66.4	r
Kapampangan	6,827	875	r	1,000	954	0.2/7	r
Urdu	7,674	2,163	u	36,428	30,784	2.9/57.1	**r**
Irish	9,859	2,183	r	887	869	0.2/23.5	r
Azeri	12,568	2,518	r	231,891	49,396	0.6/42.7	r
Tamil	13,470	3,484	m	165,004	68,980	2.6/43.6	m
Albanian	13,714	3,197	r	188,563	124,289	5.4/55.6	r
Afrikaans	14,315	4,637	r	11,389	9,861	0.8/50.2	r
Hindi	14,824	5,349	d	58,179	58,179	2.2/37.7	**r**
Bangla	16,026	2,607	b	1,606	1,280	0.2/20.6	b/**r**
Tagalog	17,757	2,534	r	247,662	124,046	3.5/65.8	r
Latvian	22,737	5,064	r	148,363	123,160	3.9/65.5	r
Bosnian	23,144	5,457	r	18,283	16,198	1/44.5	r
Welsh	25,292	3,592	r	25,832	22,280	2/50.1	r
Latin	31,195	3,380	r	18,884	11,645	0.9/36.3	r
Basque	38,594	6,058	r	880	793	0.1/7	r
Thai	40,182	5,544	t	14,925	6,054	0.5/36.4	t
Farsi	58,651	12,291	a	198,605	81,988	2.6/60.5	a/**r**
Bulgarian	68,446	19,045	c	316,631	220,829	3/66.3	c/**r**
Serbian	71,018	20,083	c	168,140	145,297	3.7/71.2	**r**
Indonesian	73,962	18,021	r	67,633	30,491	1.1/61.5	r
Slovak	76,421	15,341	r	233,093	187,234	2.6/65	r
Korean	84,385	18,638	k	229,742	120,666	2.2/66.5	k
Turkish	86,277	22,080	r	1,272,881	681,172	3.3/47.9	r
Ukrainan	91,022	22,383	c	14,056	14,056	0.5/35.3	**r**
Romanian	97,351	21,157	r	249,479	197,234	2.4/64.3	r
Russian	295,944	105,084	c	423,009	262,675	1.6/57.5	c
Spanish	371,130	189,171	r	347,441	283,138	1.7/54.7	r
Polish	438,053	96,739	r	261,463	220,119	1.3/57.8	r

Table 1: Wikipedia and dictionary statistics. The third and last columns contain predominant scripts in Wikipedia and dictionary scripts: Roman (r), Cyrillic (c), Arabic (a), Korean (k), Thai (t), Bangla (b), Devanagari (d), Tamil (m), Urdu (u), Indic (i), or other (o). Romanized dictionaries are marked in bold. In this work, we only use single token translations (6th column) for projecting contextual vectors and evaluation.

Language	Days with data											Total
	'00	'01	'02	'03	'04	'05	'06	'07	'08	'09	'10	tokens
Kyrgyz	-	-	-	-	-	-	1	7	5	15	40	126,217
Somali	-	-	-	-	-	-	-	-	28	113	64	421,589
Nepali	-	-	-	-	1	3	4	2	6	13	8	59,309
Kazakh	-	-	-	-	-	-	-	-	1	45	32	1,839,748
Uzbek	-	-	-	-	-	-	22	29	79	49	33	507,320
Urdu	-	-	1	44	33	8	36	30	76	281	98	2,874,969
Tamil	-	-	-	-	-	1	1	1	13	28	27	61,990
Albanian	1	-	-	-	2	26	13	98	94	120	70	828,005
Hindi	-	14	181	285	283	332	334	325	331	349	102	12,589,950
Bangla	-	-	-	-	-	1	1	-	4	18	29	129,984
Latvian	-	1	1	1	1	1	3	21	178	341	75	28,309,813
Bosnian	-	-	-	-	-	-	1	-	-	-	41	79,206
Welsh	-	2	-	142	245	129	1	1	1	5	-	2,603,551
Farsi	-	10	49	114	160	301	297	339	366	365	109	26,041,987
Serbian	-	1	1	-	-	-	3	-	14	84	19	221,503
Indonesian	-	-	-	-	-	-	-	1	1	59	99	1,135,783
Slovak	1	-	-	-	1	1	-	158	356	364	100	103,732,925
Turkish	-	-	-	1	7	12	8	9	24	153	69	1,135,200
Ukrainan	-	-	-	15	26	30	70	89	164	102	76	1,254,852
Russian	46	365	363	353	350	353	353	358	365	365	100	47,857,954
Spanish	-	313	352	364	366	365	365	365	366	365	104	59,732,042
English	366	365	365	365	366	365	365	365	366	365	187	1,090,171,115
Arabic	-	-	-	-	-	-	-	-	-	144	100	1,189,680
Pashto	-	-	-	-	-	3	10	9	7	26	37	520,450
Chinese	-	-	-	-	-	5	1	6	7	14	32	1,864,565

Table 2: A fraction of crawled newswire pages we have gathered for which we inferred both language and publication date information.

Parallel texts. Europarl [Koehn, 2005] is a parallel corpus in 11 languages compiled from European parliament proceedings published on the web. Besides sentence alignment, a portion of the corpus contains temporal annotation. While we cannot assume such resources to be available for all of our language pairs, we use it to get a sense of an upper bound of the performance we can achieve with our methods.

5 Experimental Evaluation

We evaluate performance of the lexicon induction framework on the monolingual resources we have collected and described in Section 4. In the first set of experiments, we consider a high resource language pair to establish relative efficacy of the cues and to get a sense of an upper bound on the overall performance. We then induce translations between English and ten low resource languages. Since many of the seed and test dictionaries are sparse and noisy, we investigate the feasibility of crowd-sourcing annotations and their use in an iterative induction procedure.

For each language, we construct a test dictionary by randomly selecting 10% of the entries in the corresponding bilingual dictionary (see Section 4), and leave the remainder as a seed lexicon for the contextual cue. We select 1000 source words from the test dictionary and compute the accuracy of the induced translations. The performance is measured by top-k accuracy, i.e. the proportion of the source words which have at least one test dictionary translation among top k of its induced candidates. More formally, denote D a set of tuples $\{(s_i, l_i)\}_{i=1}^{1000}$ of source words and their corresponding ranked lists of translations, a function $r(l, j)$ which returns a set of translation candidates at position j or above in ranking l, and a function $d(s, w)$ which returns true iff any translation candidates in a set w for s is present in the test dictionary. The accuracy at top-k is then computed as follows[2]:

$$g(U, k) = \frac{1}{|U|} \sum_{(s,t) \in U} [\![d(s, r(l, k))]\!] \tag{2}$$

Note that the reported performance is conservative since (1) as we discussed, our test dictionaries are both noisy and sparse, and (2) $g(U, k)$ does not account for multiple correct translations among top k candidates. Unless mentioned otherwise, we use the simple equivalence class heuristic (i.e. each unique token is assigned its own equivalence class) when constructing both contextual vectors and candidates. We do not consider stop words when constructing a list of source words and potential target translations. For languages without a stop word list, we drop 200 equivalence classes which occur most frequently in our corpus. However, we do not perform this pruning step when collecting contextual vectors.

As we have pointed out, one of the objectives of this work is to induce lexicons with wide coverage. Thus, unlike most of the previous work, we are particularly interested in how well we can do on inducing translations for source equivalence classes selected *randomly* from the test dictionary.

5.1 Lexicon induction for a high-resource language pair

We begin by studying some of the monolingual cues we have introduced in Section 3 as well as the MRR rank aggregation scheme on a high resource language pair. Besides providing a sense of an upper bound of performance of the individual cues, this setting is convenient for testing aggregation strategies, similarity metrics, etc. Figure 4 plots the relative performance of contextual, temporal, and orthographic cues on the English-Spanish part of the Europarl parallel data for most frequent and random 1000 words source present in our test dictionary (top left and right, respectively). The corpus is time stamped (spanning 656 days) and perfectly *temporally* aligned, so it is not surprising that the temporal cue provides a strong signal for inducing translations. All of these cues are informative and orthogonal, so combining them substantially improves results

[2] $[\![\cdot]\!]$ is one if the predicate inside the brackets is true, and zero otherwise.

with 73% and 82% accuracy at top 100 and 500 for most frequent source words, and 70% and 79%, respectively, for random source tokens.

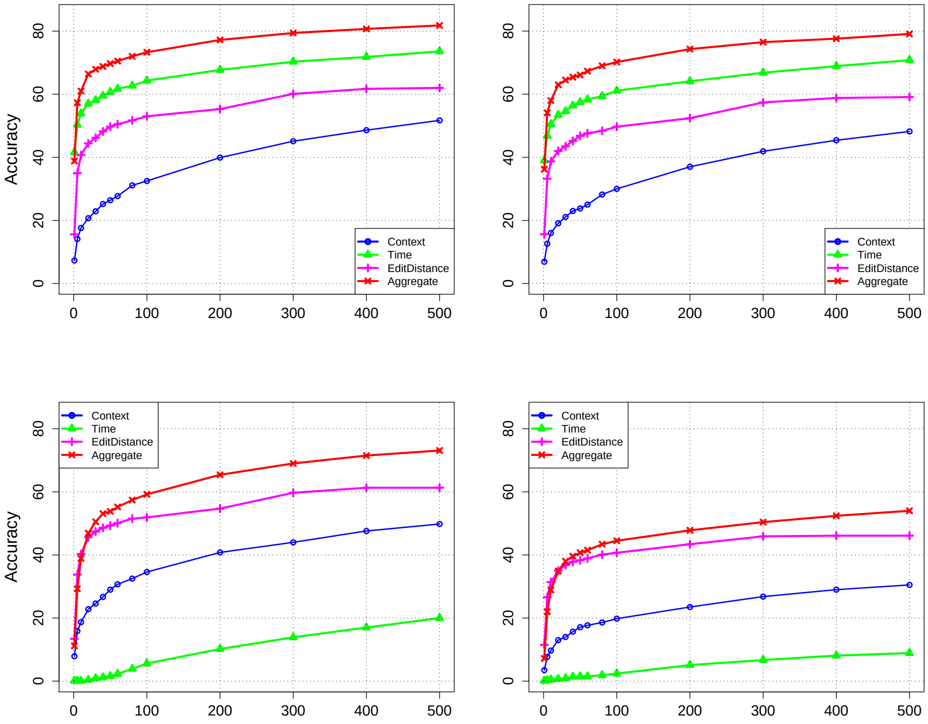

Figure 4: Accuracy on parallel en-es data for most frequent (top left) and random (top right) 1000 source words, and on wikipedia / newswire for most frequent (bottom left) and random (bottom right) 1000 source words.

We repeat the same experiment using wikipedia and newswire data to derive dictionaries from contextual and temporal cues, respectively (Figure 4, bottom). Since the newswire corpora is only weakly temporally aligned, the performance of the temporal cue drops substantially. Still, the cues provide informative and non-redundant signals, which can be combined to obtain better quality lexicons. Notice the substantial drop in performance (about **19%**) for randomly selected source tokens. While not entirely surprising, it supports our observation that evaluating on frequent source words alone can be misleading if the objective is to induce wide coverage dictionaries.

Table 3 contains ranked lists of translations induced from the three cues and aggregated using the MRR heuristic for three sample english words: au-

cues			
contextual	temporal	orthographic	MRR
sadam	**autoritario**	*autoritaria*, *autoritarias*	**autoritario**
autoritario	democracia	**autoritario**, *autoritarios*	sadam
azúcar	*autoritarios*	autoritarismo, ...	*autoritaria*
perfeccionamiento	pas	autorizadas, ...	*autoritarias*
tránsito	democráticas	horarias, ...	*autoritarios*
apartheid	elecciones	autocar comisaria ...	democracia
transitorio	fuerzas		azúcar
totalitario	vecino		perfeccionamiento, pas
talibán	democráticos		democráticas, tránsito
dictatorial	*autoritaria*		elecciones, apartheid
infantil	*lactantes*	**infantil**, infame, ...	**infantil**
materna	mortalidad	infames, infamia, ...	*lactantes*
maternal	sociedad	isaf, influir, ...	materna, mortalidad, ...
morbilidad	nivel		maternal, sociedad
ndice	problema		morbilidad, nivel
tasa	trabajo		ndice, problema
disminuira	pases		tasa, trabajo
crónico	tiempo		pases, disminuira
escape	reducción		crónico, tiempo
derivada	desarrollo		reducción, escape
almacenamiento	**almacenamiento**	jorge, seora, ...	**almacenamiento**
almacenaje	nuclear	saturado, gestoras, ...	**almacenaje**, nuclear
recoger	radiactivos		recoger, radiactivos
médico	almacenar		almacenar, médico
tratamiento	radioactivos		tratamiento, radioactivos
servicio	propuesta		propuesta, servicio
viaje	normas		viaje, normas
vuelo	peligro		vuelo, peligro
registro	llegar		llegar, registro
etc	instalaciones		instalaciones, etc

Table 3: Top ranked translations for **authoritarian**, **infant**, **storage** (top, middle, bottom, respectively) inferred from context, temporal, and orthographic cues and aggregated with the MRR heuristic. Correct translations are in bold if found in the test dictionary and italicized otherwise.

thoritarian, infant, and storage. Note the numerous morphological variants of the correct translations and alternative translations not found in the dictionary (italicized). Also notice that when some signals are not informative (e.g. the orthographic cue in the third example) others may still be sufficient to induce the correct candidates at the top of the aggregated list.

5.2 Context vs. alignments

Word alignments can be induced directly from sentence aligned corpora. In this set of experiments, we compare bilingual lexicons derived from word alignments (produced by GIZA++, [Och and Ney, 2003]) to those generated from the contextual cue alone on the English-Spanish section of the Europarl corpus (see Figure 5). While word alignments induce a much more informative signal than context alone, sufficient sentence aligned bilingual data is not available for most of the low resource languages we consider in this work.

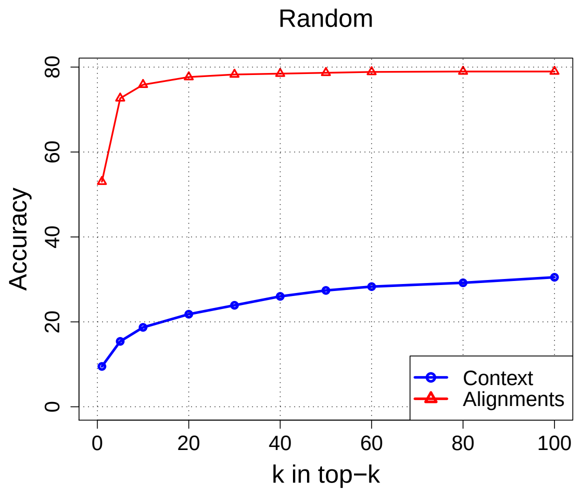

Figure 5: Accuracy of lexicons induced from alignments and context on parallel en-es data for random 1000 source words.

5.3 Lexicon induction for a low-resource languages

Next, we selected 10 low resource languages for which we have collected sufficient amount of newswire data and have seed dictionaries in an appropriate script: Welsh, Farsi, Indonesian, Latvian, Russian, Slovak, Albanian, Serbian, Turkish, and Uzbek. Figure 6 shows the accuracy of lexicons induced from the contextual (top), and temporal (middle) cues, and the MRR aggregation scheme (bottom); results for 1000 most frequent and random source words are shown on the left and right, respectively.

Similarly to the experiments in Section 5.1, lexicon accuracy drops on random source words for Russian and Turkish. For the remaining languages, low dictionary coverage means a large overlap between 1000 random and most frequent test dictionary terms. That is, both sets of experiments effectively test the accuracy on random source words, since test dictionary entries themselves were selected at random. Both temporal and contextual cues are informative, but the simple aggregation heuristic fails to provide an improvement. While simple ensemble schemes typically work well for a large number of diverse signals, a more sophisticated approach to combining few available ranked lists remains the subject of our ongoing work.

Confusing? Drop most frequent?

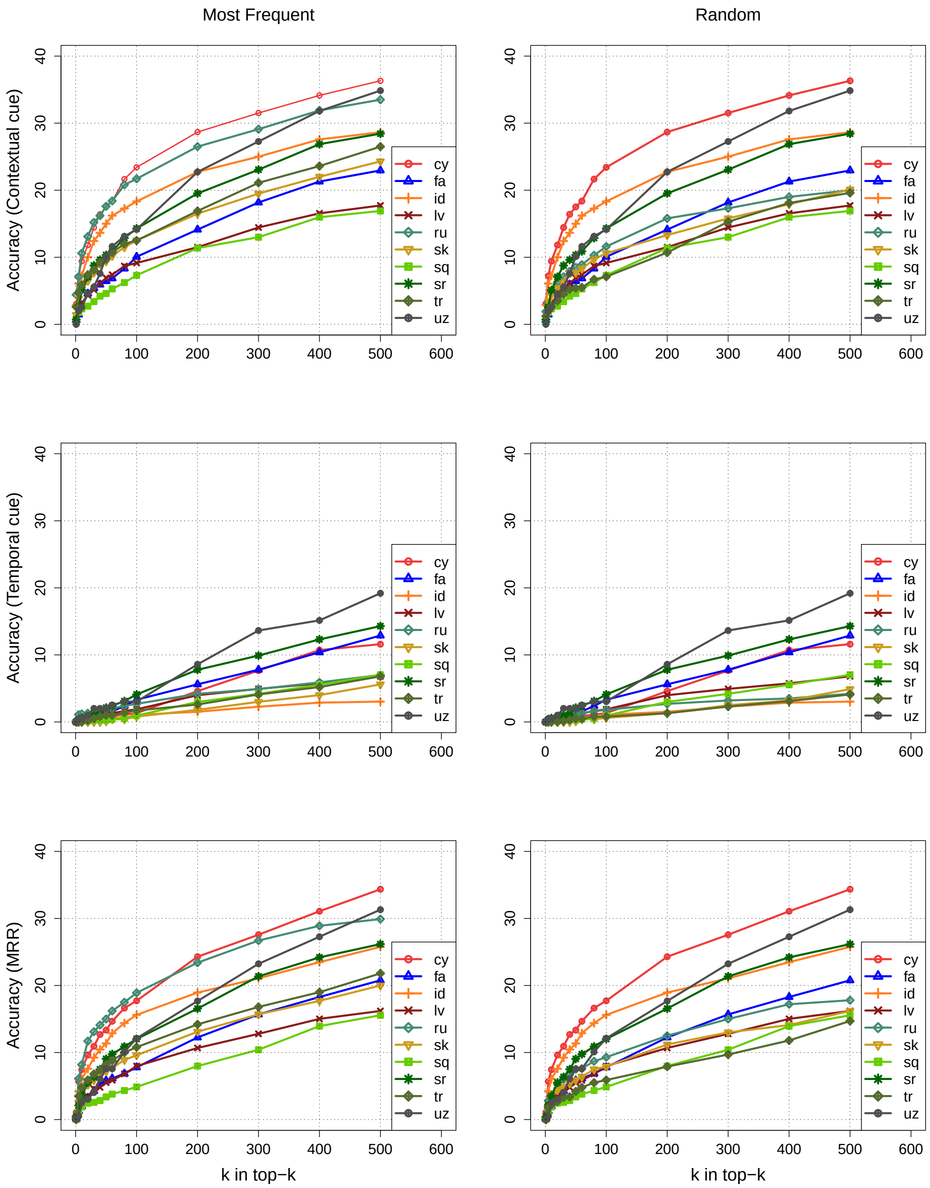

Figure 6: Accuracy of contextual (top) and temporal (middle) cues and the MRR scheme (bottom) for most frequent (left) and random (right) 1000 source words for Welsh (cy), Farsi (fa), Indonesian (id), Latvian (lv), Russian (ru), Slovak (sk), Albanian (sq), Serbian (sr), Turkish (tr), and Uzbek (uz).

5.4 Crowd-sourcing annotation

We use our existing bilingual dictionaries to induce large bilingual lexicons via the contextual cue and to evaluate their accuracy. However, these dictionaries vary substantially in quality and coverage across languages and corpora (see Section 4). In this set of experiments we follow previous work on crowd-sourcing annotations [Snow et al., 2008, Callison-Burch, 2009] and study [Irvine and Klementiev, 2010] the viability of obtaining translations for a low-resource languages specifically for use in our induction framework. First, we use the contextual cue to induce lexical translation pairs from the Wikipedia monolingual data. Then, we pay Amazon Mechanical Turk (MTurk) workers a small amount to check and correct our system output. We can then use the updated lexicons to inform another iteration of lexicon induction, gather a second set of MTurk annotations, and so on.

For 32 of the 42 languages in Table 1, we were able to induce lexical translation candidates and post them on MTurk for annotation. For these languages we presented annotators with top ten scored candidates for a set of 100 English words and asked them to mark correct translations. If our seed dictionary included an entry for a source word, we included the translation in the candidate list as a positive control. Additionally, we included a random word in the foreign language as a negative control. We do not have dictionaries for the remaining 10 languages, so we asked workers to type translations for 100 English words. We had three distinct workers provide such annotations for each source word.

Figure 7 (top) shows the time it took to complete annotation of for 37 languages on MTurk. Annotations the following languages were posted for a week and were never completed: Tigrinya, Uighur, Tibetan, Kyrgyz, and Kazakh. All five of the uncompleted required typing annotations, a more time consuming task than checking translation candidates. Not surprisingly, languages with many speakers (Hindi, Spanish, and Russian) and languages spoken in and near India (Hindi, Tamil, Urdu) were completed very quickly. Figure 7 (bottom) shows the percent of positive control candidate translations that were checked by the majority of workers (at least two of three). The highest amounts of agreement with the controls were for Spanish and Polish, which indicates that those workers completed the annotations more accurately than the workers who completed, for example, the Tatar and Thai annotations. However, the seed dictionaries are very noisy, so this finding may be confounded by discrepancies in the quality of our dictionaries. The noisy dictionaries also explain why agreement with the positive controls is, in general, relatively low.

To understand the utility of MTurk generated translation for inducing lexicons, we supplemented our dictionaries for each of the 37 languages for which we gathered MTurk annotations with translation pairs that workers agreed were good (both chosen from the candidate set and manually translated). We compared seed dictionaries of size 200 with those supplemented with, on average, 69 translation pairs. We found an average relative increase in accuracy of our output candidate set (evaluated against complete available dictionaries) of **53%**.

In sum, we found that the iterative approach of automatically generating

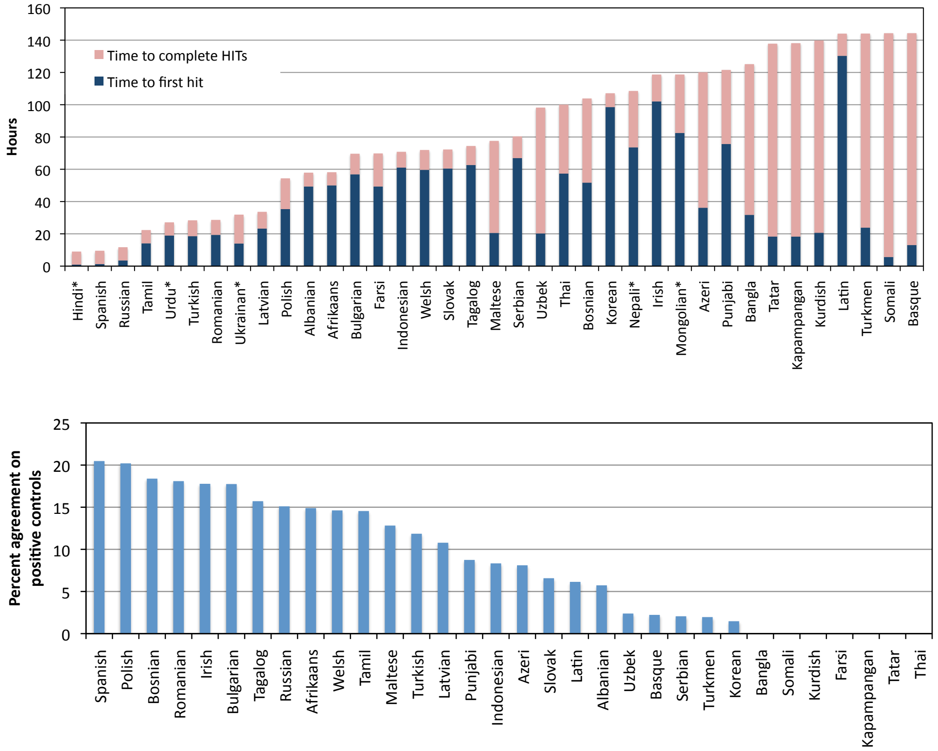

Figure 7: Top: time to complete annotation of 100 English words. Division of the time between posting and the completion of the first annotation unit (HIT) and the time between the completion of the first and last HIT shown. HITs that required lexical translation only are marked with an $*$. Bottom: percent of positive control candidate translations for which two or three workers checked as accurate.

noisy annotation and asking MTurk users to correct it to be an effective means of obtaining supervision. These correction tasks are simple, can be completed quickly for a large number of low resource languages, and produce high quality annotation.

6 System Overview

In this section we touch on some of the implementation details of the lexicon induction framework: we overview the data collection and lexicon induction procedures and explain how the framework can be extended to include new monolingual resources and cues derived from them.

6.1 Data collection

While some monolingual resources (see Section 4) are static, others require ongoing collection. We have set up the nutch crawler[3] to continuously crawl a set of web sites generating news content in the languages of interest (see Table 1 for the current summary of the collected data). The crawl results are periodically processed (see Figure 8) by the following steps implemented as Hadoop MapReduce jobs:

1. Parse new page content and extract metadata associated with each page (see `NutchPageExtractor`).
2. Merge new and previously extracted pages (`PageMerger`).
3. For each new page, identify the content language either by using the Google Language API (`LangIdentifier`) or from its URL using site-specific heuristics (`URLAndContentsLangTimeExtractor`). Identify publication time either from page content or its URL using site-specific heuristics.
4. Generate a dataset for each language from pages annotated withe language and time in the previous step (`DatedCorpusGenerator`).

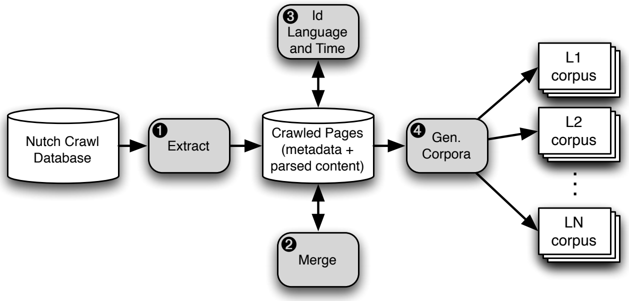

Figure 8: Ongoing data collection.

Including a new language in the collection process is simple. First, a web site publishing content in that language is added to a list of pages crawled by the nutch crawler. Then, a set of heuristics to identify language and publication time from page URL, content, or metadata collected by the crawler is added to `URLAndContentsLangTimeExtractor`.

6.2 Lexicon induction procedure

We now briefly describe how the procedure for inducing a bilingual dictionary for a given set of words is implemented in code. We highlight some of the most relevant classes, and show how the framework can be extended to include new monolingual

[3] http://nutch.apache.org/

resources and cues. Figure 9 gives a high level view of the lexicon induction procedure.

We argued in Section 3.3 that collecting aggregate statistics for morphological variants of a single lexeme is important when dealing with morphologically rich languages. Base class `EquivalenceClass` groups morphological variants present in the data into equivalence classes and maintains a set of aggregate statistics derived from monolingual cues. In turn, each of the statistics, or properties, is implemented by a subclass of `Property`.

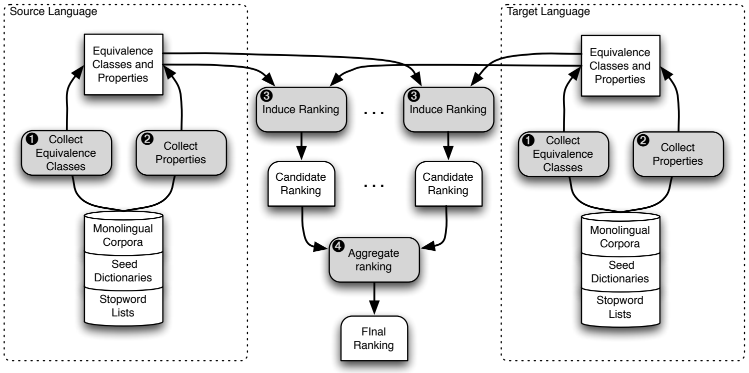

Figure 9: A high level overview of the lexicon induction framework. Equivalence classes and corresponding properties are first extracted from monolingual data (steps 1 and 2). Similarity metrics defined over the properties are then used top produce rankings over target candidates in step 3. Finally, ranked lists are aggregated to produce the final rankings in step 4.

The induction procedure begins with two passes through both source and target language monolingual corpora (step 1 and 2 on Figure 9, respectively) implemented in `DataPreparer`. Each of the available corpora (e.g. see Section 4) is accessed through a corresponding subclass of `CorpusAccessor`. In the first pass, morphological variants are collected (see `EquivalenceClassCollector`) to generate the corresponding equivalence classes and, in the second pass, a set of properties such as contextual vectors, temporal distributions, topic information, etc. is collected for each of the equivalence classes. The initial set of equivalence classes is pruned by a series of filters extending `EquivalenceClassFilter` in order to throw out patently incorrect or undesirable candidates, e.g. stop words, least or most frequent classes, strings containing numbers or letters of a wrong script, etc. Both source and target equivalence classes along with the collected statistics are persisted on disk.

Next, collected properties along with the corresponding similarity metrics extending `Scorer` are used to produce a ranked list of candidates for each of the source equivalence classes. This step involves a substantial amount of computation since each of the source equivalence classes is compared with all of the target candidates. Its implementation in `Ranker` is parallelized, which substantially speeds up this step. Ranked candidate lists induced for each source equivalence class from multiple cues are aggregated (see `Reranker`) into a joint ranking in step 4 on Figure 9 Finally, the induced ranked candidate lists are evaluated in `NBestCollector`.

Listing 1 shows an example configuration file for setting up the induction process. It is split into 5 sections, one would:

- The `corpora` section lists both source and target monolingual corpora with additional configuration parameters specific to the corresponding subclass of `CorpusAccessor`.
- The `resources` section specifies additional resources, such as stop word lists and bilingual dictionaries.
- The `preprocessing` section configures the two stage preprocessing stage, i.e. which resources to use to generate equivalence classes and how to collect their properties. For example, the `candidates` section on Listing 1 specifies that the simple and prefix heuristics (see Section 3.3) should be used for generating source and target equivalence classes, respectively, and that the classes should be pruned if they occur fewer than 10 times in the data.
- Finally, the `experiments` section configures the induction process. The configuration parameters can be used to choose most frequent or random source equivalence classes for induction (`RandomSource`), the portion of the dictionary to use for projecting contextual vectors (`DictionaryPercentToUse`), the target candidate ranked lists size to induce (`NumTranslationsToAddPerSource`), the number of threads to use when generating rankings (`NumRankingThreads`), and to specify which properties are to be used to induce those rankings and whether or not to aggregate them (`DoAggregate`).

In order to extend the framework to add a new monolingual resource and/or include additional cues:

- Extend `CorpusAccessor` to enable access to a new resource.
- Extend `Property` and `PropertyCollector` to manage and collect desired statistics from a monolingual resource.
- Extend `Scorer` to implement a similarity metric for scoring a source and a target candidate equivalence classes.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<configuration>
<corpora>
  <wiki>
    <Path>./resources/wiki/en-ru</Path>
    <SrcRegExp>.*\\.en</SrcRegExp>
    <TrgRegExp>.*\\.ru</TrgRegExp>
  </wiki>
  <crawls>
    <Path> ./resources/crawls</Path>
    <SrcSubDir>en/</SrcSubDir>
    <TrgSubDir>ru/</TrgSubDir>
    <DateFrom>00-01-01</DateFrom>
    <DateTo>10-04-20</DateTo>
  </crawls>
</corpora>
```

```
<resources>
  <stopwords>
    <Path> ./resources/stopwords/</Path>
    <SrcStopWords>en.stop</SrcStopWords>
    <TrgStopWords>ru.stop</TrgStopWords>
  </stopwords>
  <dictionary>
    <Path> ./resources/dictionaries/</Path>
    <Dictionary>en-ru.dict</Dictionary>
  </dictionary>
</resources>
<preprocessing>
  <Path>./preprocessing/</Path>
  <FilterRomanTrg>false</FilterRomanTrg>
  <input>
    <Context>wiki</Context>
    <Time>crawls</Time>
  </input>
  <candidates>
    <SrcEqClass>babel.content.eqclasses.SimpleEquivalenceClass</
        SrcEqClass>
    <TrgEqClass>babel.content.eqclasses.PrefixEquivalenceClass</
        TrgEqClass>
    <PruneIfOccursMoreThan>-1</PruneIfOccursMoreThan>
    <PruneIfOccursFewerThan>10</PruneIfOccursFewerThan>
    <PruneMostFrequentSrc>-1</PruneMostFrequentSrc>
    <PruneMostFrequentTrg>-1</PruneMostFrequentTrg>
  </candidates>
  <context>
    <SrcEqClass>babel.content.eqclasses.SimpleEquivalenceClass</
        SrcEqClass>
    <TrgEqClass>babel.content.eqclasses.SimpleEquivalenceClass</
        TrgEqClass>
    <PruneEqIfOccursMoreThan>-1</PruneEqIfOccursMoreThan>
    <PruneEqIfOccursFewerThan>5</PruneEqIfOccursFewerThan>
    <PruneContextToSize>-1</PruneContextToSize>
    <Window>2</Window>
  </context>
  <time>
    <Align>true</Align>
  </time>
</preprocessing>
<output>
  <Path>./output/</Path>
</output>
<experiments>
  <time>
    <SlidingWindow>false</SlidingWindow>
    <WindowSize>1</WindowSize>
  </time>
  <RandomSource>false</RandomSource>
  <NumSource>1000</NumSource>
  <NumTranslationsToAddPerSource>500</NumTranslationsToAddPerSource
      >
  <DictionaryPercentToUse>0.9</DictionaryPercentToUse>
  <DictionaryPruneNumTranslations>-1</
      DictionaryPruneNumTranslations>
  <NumRankingThreads>15</NumRankingThreads>
  <DoTime>true</DoTime>
  <DoContext>true</DoContext>
  <DoEditDistance>false</DoEditDistance>
  <DoAggregate>true</DoAggregate>
</experiments>
</configuration>
```

Listing 1: Example configuration file.

7 Challenges

As quantities of data we collect continues to increase, so does the computational cost associated with processing them. When searching for translations, we consider each of the M target candidates for all of the N source words making $M \times N$ similarity computations for each cue. Growing monolingual datasets mean larger sizes of both source and target vocabularies due to open-class words (e.g. Named Entities, slang, borrowed words, etc.). Moreover, vector representations of both source words and target candidates also grow (e.g. larger context vectors and temporal distributions) requiring more cycles for each similarity computation. Needless to say, the computational costs jump substantially as we move from word to phrase translation lexicons. To address this challenge, we are building on existing work (e.g. [Li et al., 2008, Li and König, 2010, Van Durme and Lall, 2009a, Van Durme and Lall, 2009b, Van Durme and Lall, 2010]) to efficiently compute token co-occurrence metrics over large document collections. Using these techniques will enable us to process extremely large data improving both the coverage and quality of induced lexicons.

Noisy dictionaries mean less informative contextual cue as well as inaccurate evaluation of induced lexicons. In Section 5.4 we showed that crowed-sourced annotation can be obtained cheaply for most of the low resource languages we considered. We plan to continue investigating methods for gathering human annotation for both induction and evaluation.

Simple heuristics for handling morphology (see Section 3.3) may be sufficient for some of the languages we consider. However, improving lexicons for morphologically productive languages may require more sophisticated techniques. We cannot generally assume that we have access to linguistic expertise to encode morphological rules for our languages of interest, and will investigate unsupervised techniques for morphological segmentation (e.g. [Snyder and Barzilay, 2008, Poon et al., 2009]).

Aggregating multiple informative and independent cues improves the quality of induced lexicons (see Section 5.1). However, when only a few signals of varying quality are available, simple aggregation heuristics may be ineffective. We will use the available supervision (in the form of seed dictionary translations) to induce better aggregation models. For languages with insufficient dictionaries, we will consider unsupevised techniques instead (e.g. [Klementiev and Roth, 2006, Klementiev et al., 2008]).

8 Next Steps

Addressing the challenges outlined in the previous section should improve the preliminary lexicon induction results we described in Section 5. However, one of the main objectives of the BABEL project and our next immediate goal is to study machine translation models that can be trained on these resources instead of exclusively relying on parallel data.

Current state of the state of the art translation systems (e.g. [Chiang, 2005]) use the log-linear formulation [Och and Ney, 2002] to model translation:

$$P(\mathbf{e}|\mathbf{f}) \propto \exp \sum_{i=1}^{n} \lambda_i h_i(\mathbf{e}, \mathbf{f}) \tag{3}$$

where h_i is an arbitrary feature function that can be used to score a hypothesis English translation **e** of some foreign sentence **f**. Feature functions typically include a phrase translation probability, a language model probability, a lexical translation probability, and a reordering probability. The weights λ_i are generally estimated to minimize the amount of translation error against a known reference translation. We will use this discriminative formulation to model translation for low-resource languages, but will augment it with feature functions derived from monolingual cues we have defined.

Our immediate goal is to study how informative the monolingual data we have collected are for inferring phrase translation probabilities when varying amount of parallel data is available:

- First, we will assume that phrase pairs are extracted from a full parallel corpus for a resource-rich language pair. We will include the feature functions derived from monolingual cues to see if any improvement can be made in the high resource setting.
- Varying the amount of parallel data for estimating phrase translation probabilities should provide the first indication about the tradeoff between the monolingual and parallel data resources required to achieve similar performance.
- When no parallel data is available, monolingual data itself must be used to derive phrase pairs as well as score their translations. Successfully extracting a translation table requires addressing the computational challenge outlined in the previous section.
- Extending the results to a large set of low resource languages is the ultimate goal of this stage of the projects.

After the lexical choice, the next most complicated aspect of machine translation is placing translated words in correct order. While n-gram language models constraint word order, an explicit estimate of phrase reordering probability has a significant impact on the quality of resulting translation. Inducing these probabilities from monolingual texts is the next goal of the project. We believe that we can induce these probabilities by looking at pairs of adjacent words or phrases, and seeing how often their known translations co-occur in sentences in monolingual target language texts.

References

[Callison-Burch, 2009] Callison-Burch, C. (2009). Fast, cheap, and creative: Evaluating translation quality using amazons mechanical turk. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

[Chiang, 2005] Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL).*

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

[Fung and Yee, 1998] Fung, P. and Yee, L. Y. (1998). An ir approach for translating new words from nonparallel, comparable texts. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 414–420.

[Garera et al., 2009] Garera, N., Callison-Burch, C., and Yarowsky, D. (2009). Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL).*

[Haghighi et al., 2008] Haghighi, A., Liang, P., Berg-Kirkpatrick, T., and Klein, D. (2008). Learning bilingual lexicons from monolingual corpora. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL).*

[Hetland, 2004] Hetland, M. L. (2004). A survey of recent methods for efficient retrieval of similar time sequences. In *Data Mining in Time Series Databases*, chapter 2. World Scientific.

[Irvine and Klementiev, 2010] Irvine, A. and Klementiev, A. (2010). Using mechanical turk to annotate lexicons for less commonly used languages. In *The NAACL Workshop on Creating Speech and Language Data With Amazon's Mechanical Turk.*

[Klementiev and Roth, 2006] Klementiev, A. and Roth, D. (2006). Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL).*

[Klementiev et al., 2008] Klementiev, A., Roth, D., and Small, K. (2008). Unsupervised rank aggregation with distance-based models. In *Proc. of the International Conference on Machine Learning (ICML).*

[Koehn, 2005] Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proc. of the Machine Translation Summit.*

[Koehn and Knight, 2002] Koehn, P. and Knight, K. (2002). Learning a translation lexicon from monolingual corpora. In *ACL Workshop on Unsupervised Lexical Acquisition.*

[Li et al., 2008] Li, P., Church, K. W., and Hastie, T. J. (2008). One Sketch For All: Theory and Application of Conditional Random Sampling. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS).*

[Li and König, 2010] Li, P. and König, A. C. (2010). b-Bit minwise hashing. In *Proc. of the International Conference on World Wide Web (WWW).*

[Mann and Yarowsky, 2001] Mann, G. S. and Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

[Mimno et al., 2009] Mimno, D., Wallach, H., Naradowsky, J., Smith, D., and McCallum, A. (2009). Polylingual topic models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Och and Ney, 2002] Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Och and Ney, 2003] Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*.

[Poon et al., 2009] Poon, H., Cherry, C., and Toutanova, K. (2009). Unsupervised morphological segmentation with log-linear models. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*.

[Rapp, 1995] Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 320–322.

[Rapp, 1999] Rapp, R. (1999). Automatic identification of word translations from unrelated english and german corpora. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 519–526.

[Schafer and Yarowsky, 2002] Schafer, C. and Yarowsky, D. (2002). Inducing translation lexicons via diverse similarity measures and bridge languages. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 146–152.

[Snow et al., 2008] Snow, R., OConnor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

[Snyder and Barzilay, 2008] Snyder, B. and Barzilay, R. (2008). Unsupervised multilingual learning for morphological segmentation. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Van Durme and Lall, 2009a] Van Durme, B. and Lall, A. (2009a). Probabilistic Counting with Randomized Storage. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[Van Durme and Lall, 2009b] Van Durme, B. and Lall, A. (2009b). Streaming Pointwise Mutual Information. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NIPS)*.

[Van Durme and Lall, 2010] Van Durme, B. and Lall, A. (2010). Online Generation of Locality Sensitive Hash Signatures. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.