Lexicon Induction for Low Resource Languages

July 16, 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 10 low resource languages. Since our principal objective is to induce lexicons with broad coverage, 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.

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 those less frequently taught.

On the other hand, we now have unprecedented access to vast and continually expanding monolingual resources. Moreover, they often contain additional metadata which can provide additional cues for inducing bilingual resources; suggesting we might substantially reduce and eventually eliminate the requirement for explicitly aligned bilingual translations. Recent examples include exploiting temporal information to induce Named Entity lexicons ([Schafer and Yarowsky, 2002, Klementiev and Roth, 2006]), and topic information to generate translations ([Mimno et al., 2009]).

In this work, our objective is two fold. First, we gather large amounts of cheap to collect monolingual data. Second, we exploit monolingual cues intrinsic to these resources to induce *broad coverage* bilingual translation lexicons between English and a large set of low resource languages.

We evaluate the efficacy of each of the individual cues on a language pair for which plentiful bilingual resources are available as well as each of the 12 low resource languages. Prior work typically evaluates the quality of induced lexicons on a set of words (e.g. most frequent nouns [Rapp, 1999] (add citations)) common in the source corpus. Since one of our principal objectives is to induce lexicons with broad coverage, we also evaluate the induced translations for words randomly selected out of our test dictionary.

2 Related Work

- Context: [Rapp, 1995, Rapp, 1999, Fung and Yee, 1998]
- Time: [Schafer and Yarowsky, 2002, Klementiev and Roth, 2006]
- Topics: [Mimno et al., 2009, Boyd-Graber and Blei, 2009]
- Multiple: [Schafer and Yarowsky, 2002, Koehn and Knight, 2000, Haghighi et al., 2008]
- Dependecies: [Garera et al., 2009]
- Bridge languages: [Mann and Yarowsky, 2001]
- Combination Strategies: [Koehn and Knight, 2000, Klementiev and Roth, 2006, Klementiev et al., 2008]
- Mechanical Turk: Following previous work on posting NLP tasks on MTurk [Snow et al., 2008, Callison-Burch, 2009], we use the service to gather annotations for proposed bilingual lexicon entries.
- Other: [Monz and Dorr, 2005]

3 Inducing Bilingual Lexicons From Monolingual Cues

Various linguistic and corpus cues are helpful for relating word translations across a pair of languages.

Much of the monolingual content available online contains additional metadata. News feeds, for example, are comprised of news stories annotated with date and time of publication, as well as the location and the topic(s) (e.g. sports, politics, finance, etc.) associated with the story

(What we need to do: define metric, etc...)

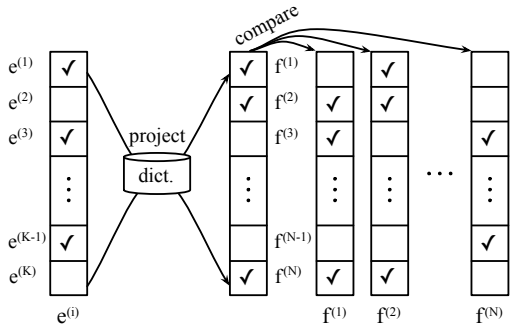

Figure 1: Lexicon induction using contextual information. First, contextual vectors are projected using small seed dictionaries and then they are compared with the target language candidates.

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 [**?**]. 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. The elements in an unknown word's vector are *projected* into the vector space of the other language using the known translations from a small seed bilingual dictionary. This sparse projected vector is compared to the vectors for all words in the target language. The word whose vector is most similar to the projected vector is considered to be the best translation of the unknown word. This process is illustrated in 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.

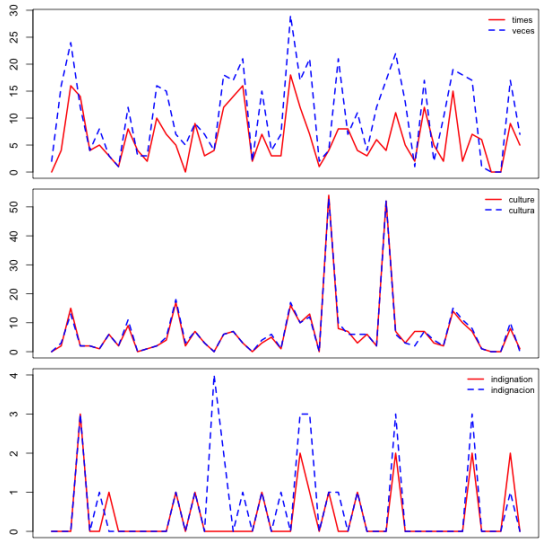

Figure 2: Temporal histograms of three English words and their Spanish translations.(From EurPoarl, change fig?)

Orthographic cue

(Edit distance)

Phonetic cue

(Named entities and cognates.)

Topic cue

(Wiki categories.)

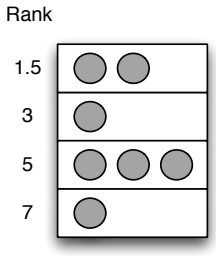

Figure 3: Resolving ties.

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.

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. [**?**]) could be incorporated instead and are a subject of ongoing work.

4 Resources

Wikipedia

(Markup removed, interlingual links used to pair up pages between English and another language. Done for 43 languages. Have topic information for the topic similarity cue. To keep them comparable, we extracted pages with interlanguage links - stats. Table 1 shows our 42 languages of interest and the number of Wikipedia articles with interlingual links to their English counterparts.)

Cannot find the coverage run logs. Compare some to Ben's runs. Check dict num tokens.

News wire

(Markup removed language identified, and temporal information extracted. Up to 10 years of temporal data for 23 languages. A substantial portion of the collected data still needs to be processed for language and time.)

(What is whown in the table **is** language id'd and time stamp extracted. LAnguage from URL, time from URL or page body. A large section of crawled data is not id'd)

50000 or more tokens, see section blah for technical details.

Note: Be's runs used an older crawl April.

Dictionaries and stopword lists

(Talk about the fact that they are noisy, romanized. Evaluation results are conservative. (Include wiki src/trg dictionary coverage, and whether or not useful directly (e.g. romanized))) Mention that if a stop word list is absent we cut out 200 most frequent words.

Dictionaries were obtained: Dictionary source: electronic dictionary (i) / ocr (o) / derived via nlp experiment (d)

We have also collected stop words for Farsi, Bangla, Hindi, Polish, Spanish, Russian, Romanian, and English. For languages which do not have a stop word lists, we remove 200 most frequent words.

Parallel texts

(MT Workshop data, some are time stamped. We use one language pair to figure an upper bound. How many days and tokens.)

Language	Wikipedia pages	Dictionary		
		entries	src token/type coverage (%)	script
Tigrinya	36	56	0.3/0.6	r
Punjabi	401	76,311	12.2/57.7	r/o
Kyrgyz	492	74,890	2.6/42.4	r
Somali	585	230	0.2/6.8	r
Nepali	1,293	6,812	3.6/39.7	r
Tibetan	1,358	59,083	2.5/35.1	r
Uighur	1,814	16,285	1.7/37.9	r
Maltese	1,896	7,574	2.9/44.7	r
Turkmen	3,137	91,928	4.6/54	r
Kazakh	3,470	145,750	1/34	r
Mongolian	4,009	948	0.3/20.7	r
Tatar	4,180	8,557	1.7/33	c/r
Kurdish	5,059	9,870	1.2/32	r
Uzbek	5,875	190,688	6.2/66.4	r
Kapampangan	6,827	1,000	0.2/7	r
Urdu	7,674	36,428	2.9/57.1	r
Irish	9,859	887	0.2/23.5	r
Azeri	12,568	231,891	0.6/42.7	r
Tamil	13,470	165,004	2.6/43.6	o
Albanian	13,714	188,563	5.4/55.6	r
Afrikaans	14,315	11,389	0.8/50.2	r
Hindi	14,824	58,179	2.2/37.7	r
Bangla	16,026	1,606	0.2/20.6	o/r
Tagalog	17,757	247,662	3.5/65.8	r
Latvian	22,737	148,363	3.9/65.5	r
Bosnian	23,144	18,283	1/44.5	r
Welsh	25,292	25,832	2/50.1	r
Latin	31,195	18,884	0.9/36.3	r
Basque	38,594	880	0.1/7	r
Thai	40,182	14,925	0.5/36.4	o
Farsi	58,651	198,605	2.6/60.5	a/r
Bulgarian	68,446	316,631	3/66.3	c/r
Serbian	71,018	168,140	3.7/71.2	r
Indonesian	73,962	67,633	1.1/61.5	r
Slovak	76,421	233,093	2.6/65	r
Korean	84,385	229,742	2.2/66.5	k
Turkish	86,277	1,272,881	3.3/47.9	r
Ukrainian	91,022	14,056	0.5/35.3	r
Romanian	97,351	249,479	2.4/64.3	r
Russian	295,944	423,009	1.6/57.5	c
Spanish	371,130	347,441	1.7/54.7	r
Polish	438,053	261,463	1.3/57.8	r

Table 1: Number of Wikipedia pages that have interlanguage links with English, along with dictionary statistics. The last column contains the dictionary script: roman (r), cyrillic (c), arabic (a), korean (k), or other (o).

Language	Pages per year											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: Crawled newswire pages with identified language and publication date information.

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 each of the low resource languages of interest. 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. Finally, we investigate heuristics we discussed in Section 3.3 for handling morphology.

In each experiment, we use 10% of the dictionary to test the induced candidate lists for 1000 source words. 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. The performance is measured by top-k accuracy, i.e. the proportion of source equivalence classes which have at least one test dictionary translation among top k of its induced candidates. Note that the reported performance is conservative since

the test dictionaries are both noisy and sparse.

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 inducing translations for source equivalence classes *randomly* selected from the test dictionary.

5.1 Lexicon induction for a high-resource language pair

We begin by evaluating the relative performance of lexicons induced from contextual, temporal, and orthographic cues as well as the MRR rank aggregation scheme on the English-Spanish EuroParl parallel data for inducing translations for most frequent and random 1000 words source present in our test dictionary (top left and right of Figure 4, 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 with 73% and 82% accuracy at top 100 and 500 for most frequent source words, and 70% and 79%, respectively, for random source tokens.

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.

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++) 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.

5.3 Lexicon induction for a low-resource languages

(Candidates and context from wiki, and time from newswire for 10 languages and English.)

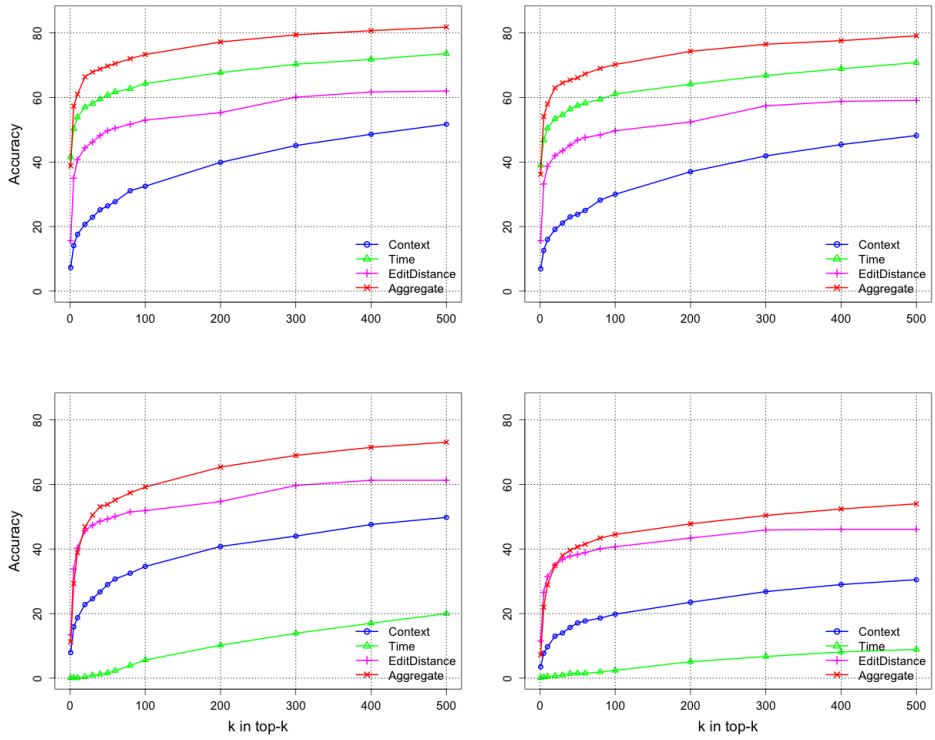

Figure 4: Accuracy on parallel en-es data for most frequent (top left) and random (top right) 1000 source words, and on wikipedia / newswire for most frequent (bottom left) and random (bottom right) 1000 source words.

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 study [Irvine and Klementiev, 2010] the viability of crown-sourcing 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

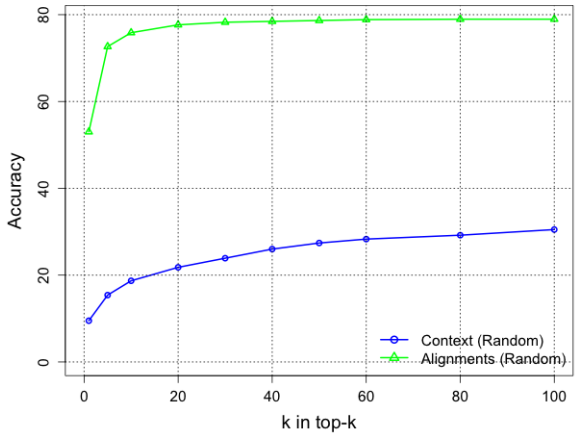

Figure 5: Accuracy of lexicons induced from alignments and context on parallel en-es data for random and most frequent 1000 source words.

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

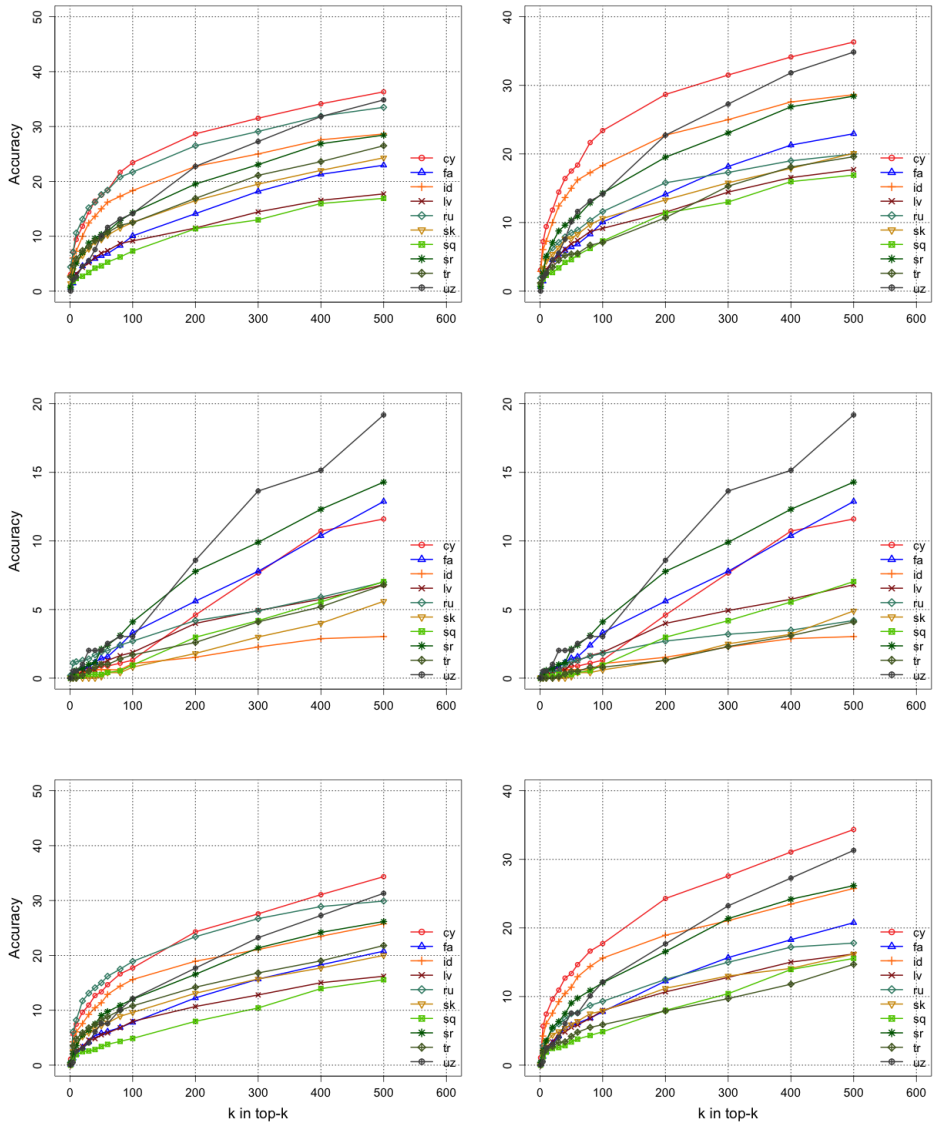

Figure 6: Accuracy of contextual (top) and temporal (middle) cues and the MRR heuristic (bottom) for most frequent (left) and random (right) 1000 source words for Welsh (cy), Farsi (fa), Indonesian (id), Latvian (lv), Russian (ru), Slovak (sk), Albanian (sq), Serbian (sr), Turkish (tr), and Uzbek (uz).

good (both chosen from the candidate set and manually translated). We compared seed dictionaries of size 200 with those supplemented with, on average,

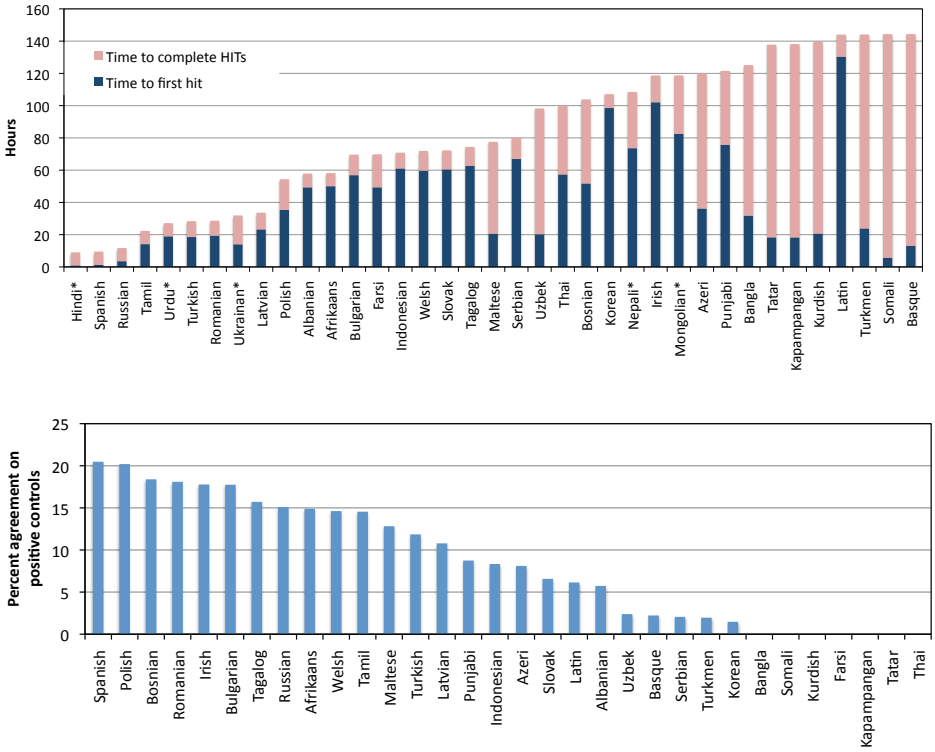

Figure 7: Top: time to complete annotation of 100 English words. Division of the time between posting and the completion of the first annotation unit (HIT) and the time between the completion of the first and last HIT shown. HITs that required lexical translation only are marked with an $*$. Bottom: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

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 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.

5.5 Dealing with morphology

(Morphological equivalence classes for context and source/target candidates.)

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[1] to continuously crawl a number of web sites generating news content in the languages of interest. The crawl results are periodically processed (see Figure 8) to (1) parse page content and extract metadata associated with the page, (2) merge with previously extracted pages, (3) identify language of the page content, and (4) generate time annotated corpora for each of the languages. See Table 1 for a current summary of the collected data.

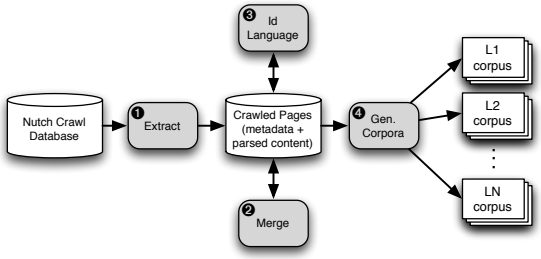

Figure 8: Ongoing data collection.

6.2 Lexicon Induction Procedure

Let us turn to the induction procedure, highlight some of the most relevant code, and show how the framework can be extended to include new monolingual resources and cues. Figure 9 shows the implementation layout and Figure 10 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`.

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

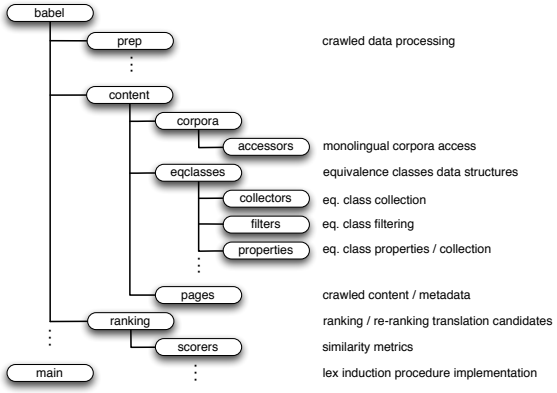

Figure 9: Implementation layout of the lexicon induction framework.

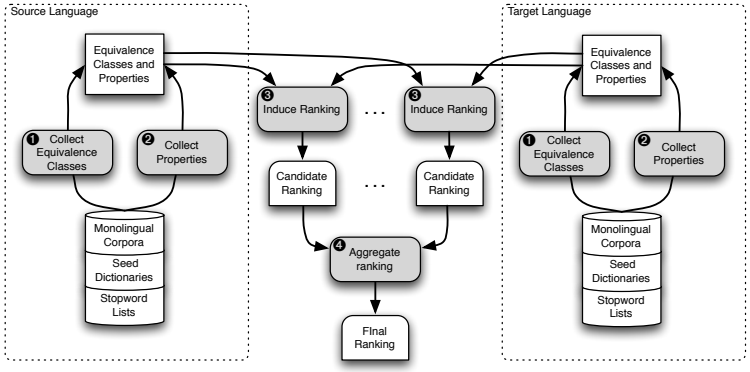

Figure 10: A high level overview of the lexicon induction framework. Equivalence classes and corresponding properties are first extracted from monolingual data (steps 1 and 2). Similarity metrics defined over the properties are then used top produce rankings over target candidates in step 3. Finally, 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 10, 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 10 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>
  <europarl>
    <Path> ./resources/es-en</Path>
    <SrcSubDir>en/</SrcSubDir>
    <TrgSubDir>es/</TrgSubDir>
    <DateFrom>96-04-15</DateFrom>
    <DateTo>06-10-13</DateTo>
  </europarl>
</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 Future Work

- MT
- Dealing with noisy dictionaries
- Better combination strategies

8 Conclusions

References

[Boyd-Graber and Blei, 2009] Boyd-Graber, J. and Blei, D. M. (2009). Multilingual topic models for unaligned text. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*.

[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)*.

[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)*.

[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 and Knight, 2000] Koehn, P. and Knight, K. (2000). Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*.

[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)*.

[Monz and Dorr, 2005] Monz, C. and Dorr, B. J. (2005). Iterative translation disambiguation for cross-language information retrieval. In *Proc. of International Conference on Research and Development in Information Retrieval (SIGIR)*.

[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)*.