Compiling a Massive, Multilingual Dictionary via Probabilistic Inference

Mausam Stephen Soderland Oren Etzioni
Daniel S. Weld Michael Skinner* Jeff Bilmes
University of Washington, Seattle *Google, Seattle
{mausam,soderlan,etzioni,weld,bilmes}@cs.washington.edu mskinner@google.com

Abstract

Can we automatically compose a large set of Wiktionaries and translation dictionaries to yield a massive, multilingual dictionary whose coverage is substantially greater than that of any of its constituent dictionaries?

The composition of multiple translation dictionaries leads to a transitive inference problem: if word A translates to word B which in turn translates to word C, what is the probability that C is a translation of A? The paper introduces a novel algorithm that solves this problem for 10,000,000 words in more than 1,000 languages. The algorithm yields PANDICTIONARY, a novel multilingual dictionary. PANDICTIONARY contains more than four times as many translations than in the largest Wiktionary at precision 0.90 and over 200,000,000 pairwise translations in over 200,000 language pairs at precision 0.8.

1 Introduction and Motivation

In the era of globalization, inter-lingual communication is becoming increasingly important. Although nearly 7,000 languages are in use today (Gordon, 2005), most language resources are mono-lingual, or bi-lingual.[1] This paper investigates whether Wiktionaries and other translation dictionaries available over the Web can be automatically composed to yield a massive, multilingual dictionary with superior coverage at comparable precision.

We describe the automatic construction of a massive multilingual translation dictionary, called PANDICTIONARY, that could serve as a resource for translation systems operating over a very broad set of language pairs. The most immediate application of PANDICTIONARY is to *lexical translation*—the translation of individual words or simple phrases (*e.g.*, "sweet potato"). Because lexical translation does not require aligned corpora as input, it is feasible for a much broader set of languages than statistical Machine Translation (SMT). Of course, lexical translation cannot replace SMT, but it is useful for several applications including translating search-engine queries, library classifications, meta-data tags,[2] and recent applications like cross-lingual image search (Etzioni et al., 2007), and enhancing multi-lingual Wikipedias (Adar et al., 2009). Furthermore, lexical translation is a valuable component in knowledge-based Machine Translation systems, *e.g.*, (Bond et al., 2005; Carbonell et al., 2006).

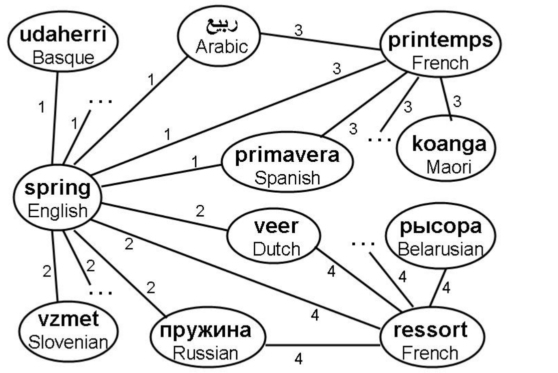

Figure 1: A fragment of the translation graph for two senses of the English word 'spring'. Edges labeled '1' and '3' are for spring in the sense of a season, and '2' and '4' are for the flexible coil sense. The graph shows translation entries from an English dictionary merged with ones from a French dictionary.

PANDICTIONARY currently contains over 200 million pairwise translations in over 200,000 language pairs at precision 0.8. It is constructed from information harvested from 631 online dictionaries and Wiktionaries. This necessitates match-

[1] The English Wiktionary, a lexical resource developed by volunteers over the Internet is one notable exception that contains translations of English words in about 500 languages.

[2] Meta-data tags appear in community Web sites such as `flickr.com` and `del.icio.us`.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 262–270,
Suntec, Singapore, 2-7 August 2009. ©2009 ACL and AFNLP

ing word senses across multiple, independently-authored dictionaries. Because of the millions of translations in the dictionaries, a feasible solution to this *sense matching* problem has to be scalable; because sense matches are imperfect and uncertain, the solution has to be probabilistic.

The core contribution of this paper is a principled method for probabilistic sense matching to *infer* lexical translations between two languages that do not share a translation dictionary. For example, our algorithm can conclude that Basque word 'udaherri' is a translation of Maori word 'koanga' in Figure 1. Our contributions are as follows:

1. We describe the design and construction of PANDICTIONARY—a novel lexical resource that spans over 200 million pairwise translations in over 200,000 language pairs at 0.8 precision, a four-fold increase when compared to the union of its input translation dictionaries.
2. We introduce SenseUniformPaths, a scalable probabilistic method, based on graph sampling, for inferring lexical translations, which finds 3.5 times more inferred translations at precison 0.9 than the previous best method.
3. We experimentally contrast PANDICTIONARY with the English Wiktionary and show that PANDICTIONARY is from 4.5 to 24 times larger depending on the desired precision.

The remainder of this paper is organized as follows. Section 2 describes our earlier work on sense matching (Etzioni et al., 2007). Section 3 describes how the PANDICTIONARY builds on and improves on their approach. Section 4 reports on our experimental results. Section 5 considers related work on lexical translation. The paper concludes in Section 6 with directions for future work.

2 Building a Translation Graph

In previous work (Etzioni et al., 2007) we introduced an approach to sense matching that is based on translation graphs (see Figure 1 for an example). Each vertex $v \in \mathcal{V}$ in the graph is an ordered pair (w, l) where w is a word in a language l. Undirected edges in the graph denote translations between words: an edge $e \in \mathcal{E}$ between (w_1, l_1) and (w_2, l_2) represents the belief that w_1 and w_2 share at least one word sense.

Construction: The Web hosts a large number of bilingual dictionaries in different languages and several Wiktionaries. Bilingual dictionaries translate words from one language to another, often without distinguishing the intended sense. For example, an Indonesian-English dictionary gives 'light' as a translation of the Indonesian word 'enteng', but does not indicate whether this means illumination, light weight, light color, or the action of lighting fire.

The Wiktionaries (wiktionary.org) are sense-distinguished, multilingual dictionaries created by volunteers collaborating over the Web. A translation graph is constructed by locating these dictionaries, parsing them into a common XML format, and adding the nodes and edges to the graph.

Figure 1 shows a fragment of a translation graph, which was constructed from two sets of translations for the word 'spring' from an English Wiktionary, and two corresponding entries from a French Wiktionary for 'printemps' (spring season) and 'ressort' (flexible spring). Translations of the season 'spring' have edges labeled with sense ID=1, the flexible coil sense has ID=2, translations of 'printemps' have ID=3, and so forth.[3]

For clarity, we show only a few of the actual vertices and edges; *e.g.*, the figure doesn't show the edge (ID=1) between 'udaherri' and 'primavera'.

Inference: In our previous system we had a simple inference procedure over translation graphs, called TRANSGRAPH, to find translations beyond those provided by any source dictionary. TRANSGRAPH searched for paths in the graph between two vertices and estimated the probability that the path maintains the same word sense along all edges in the path, even when the edges come from different dictionaries. For example, there are several paths between 'udaherri' and 'koanga' in Figure 1, but all shift from sense ID 1 to 3. The probability that the two words are translations is equivalent to the probability that IDs 1 and 3 represent the same sense.

TRANSGRAPH used two formulae to estimate these probabilities. One formula estimates the probability that two multi-lingual dictionary entries represent the same word sense, based on the proportion of overlapping translations for the two entries. For example, most of the translations of

[3] Sense-distinguished multi-lingual entries give rise to cliques all of which share a common sense ID.

French ‘printemps’ are also translations of the season sense of ‘spring’. A second formula is based on triangles in the graph (useful for bilingual dictionaries): a clique of 3 nodes with an edge between each pair of nodes. In such cases, there is a high probability that all 3 nodes share a word sense.

Critique: While TRANSGRAPH was the first to present a scalable inference method for lexical translation, it suffers from several drawbacks. Its formulae operate only on local information: pairs of senses that are adjacent in the graph or triangles. It does not incorporate evidence from longer paths when an explicit triangle is not present. Moreover, the probabilities from different paths are combined conservatively (either taking the max over all paths, or using “noisy or” on paths that are completely disjoint, except end points), thus leading to suboptimal precision/recall.

In response to this critique, the next section presents an inference algorithm, called SenseUniformPaths (SP), with substantially improved recall at equivalent precision.

3 Translation Inference Algorithms

In essence, inference over a translation graph amounts to *transitive* sense matching: if word A translates to word B, which translates in turn to word C, what is the probability that C is a translation of A? If B is polysemous then C may not share a sense with A. For example, in Figure 2(a) if A is the French word ‘ressort’ (the flexible-coil sense of spring) and B is the English word ‘spring’, then Slovenian word ‘vzmet’ may or may not be a correct translation of ‘ressort’ depending on whether the edge (B, C) denotes the flexible-coil sense of spring, the season sense, or another sense. Indeed, given only the knowledge of the path $A - B - C$ we cannot claim anything with certainty regarding A to C.

However, if A, B, and C are on a circuit that starts at A, passes through B and C and returns to A, there is a high probability that all nodes on that circuit share a common word sense, given certain restrictions that we enumerate later. Where TRANSGRAPH used evidence from circuits of length 3, we extend this to paths of arbitrary lengths.

To see how this works, let us begin with the simplest circuit, a triangle of three nodes as shown in Figure 2(b). We can be quite certain that ‘vzmet’ shares the sense of coil with both ‘spring’ and ‘ressort’. Our reasoning is as follows: even though both ‘ressort’ and ‘spring’ are polysemous they share only one sense. For a triangle to form we have two choices – (1) either ‘vzmet’ means spring coil, or (2) ‘vzmet’ means *both* the spring season and jurisdiction, but not spring coil. The latter is possible but such a coincidence is very unlikely, which is why a triangle is strong evidence for the three words to share a sense.

As an example of longer paths, our inference algorithms can conclude that in Figure 2(c), both ‘molla’ and ‘vzmet’ have the sense coil, even though no explicit triangle is present. To show this, let us define a *translation circuit* as follows:

Definition 1 *A* translation circuit from v_1^* with sense s^* *is a cycle that starts and ends at* v_1^* *with no repeated vertices (other than* v_1^* *at end points). Moreover, the path includes an edge between* v_1^* *and another vertex* v_2^* *that also has sense* s^*.

All vertices on a translation circuit are mutual translations with high probability, as in Figure 2(c). The edge from ‘spring’ indicates that ‘vzmet’ means either coil or season, while the edge from ‘ressort’ indicates that ‘molla’ means either coil or jurisdiction. The edge from ‘vzmet’ to ‘molla’ indicates that they share a sense, which will happen if all nodes share the sense season or if either ‘vzmet’ has the unlikely combination of coil and jurisdiction (or ‘molla’ has coil and season).

We also develop a mathematical model of sense-assignment to words that lets us formally prove these insights. For more details on the theory please refer to our extended version. This paper reports on our novel algorithm and experimental results.

These insights suggest a basic version of our algorithm: “given two vertices, v_1^* and v_2^*, that share a sense (say s^*) compute all translation circuits from v_1^* in the sense s^*; mark all vertices in the circuits as translations of the sense s^*”.

To implement this algorithm we need to decide whether a vertex lies on a translation circuit, which is trickier than it seems. Notice that knowing that v is connected independently to v_1^* and v_2^* doesn’t imply that there exists a translation circuit through v, because both paths may go through a common node, thus violating of the definition of translation circuit. For example, in Figure 2(d) the Catalan word ‘ploma’ has paths to both spring and ressort, but there is no translation circuit through

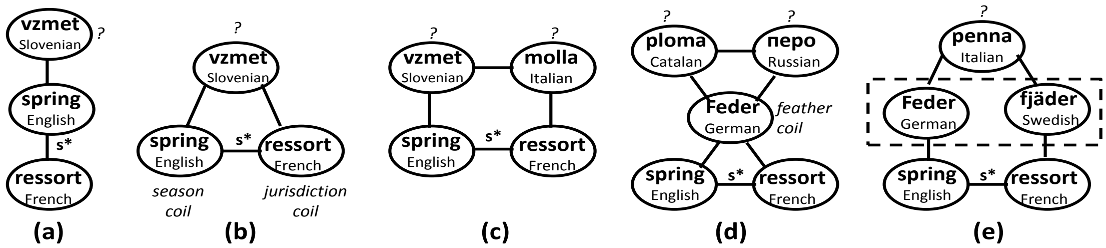

Figure 2: Snippets of translation graphs illustrating various inference scenarios. The nodes in question mark represent the nodes in focus for each illustration. For all cases we are trying to infer translations of the flexible coil sense of spring.

it. Hence, it will not be considered a translation. This example also illustrates potential errors avoided by our algorithm – here, German word 'Feder' mean feather and spring coil, but 'ploma' means feather and not the coil.

An exhaustive search to find translation circuits would be too slow, so we approximate the solution by a random walk scheme. We start the random walk from v_1^* (or v_2^*) and choose random edges without repeating any vertices in the current path. At each step we check if the current node has an edge to v_2^* (or v_1^*). If it does, then all the vertices in the current path form a translation circuit and, thus, are valid translations. We repeat this random walk many times and keep marking the nodes. In our experiments for each inference task we performed a total of 2,000 random walks (N_R in pseudo-code) of max circuit length 7. We chose these parameters based on a development set of 50 inference tasks.

Our first experiments with this basic algorithm resulted in a much higher recall than TRANSGRAPH, albeit, at a significantly lower precision. A closer examination of the results revealed two sources of error – (1) errors in source dictionary data, and (2) correlated sense shifts in translation circuits. Below we add two new features to our algorithm to deal with each of these error sources, respectively.

3.1 Errors in Source Dictionaries

In practice, source dictionaries contain mistakes and errors occur in processing the dictionaries to create the translation graph. Thus, existence of a *single* translation circuit is only limited evidence for a vertex as a translation. We wish to exploit the insight that more translation circuits constitute stronger evidence. However, the different circuits may share some edges, and thus the evidence cannot be simply the number of translation circuits.

We model the errors in dictionaries by assigning a probability less than 1.0 to each edge[4] (p_e in the pseudo-code). We assume that the probability of an edge being erroneous is independent of the rest of the graph. Thus, a translation graph with possible data errors converts into a *distribution* over accurate translation graphs.

Under this distribution, we can use the probability of existence of a translation circuit through a vertex as the probability that the vertex is a translation. This value captures our insights, since a larger number of translation circuits gives a higher probability value.

We sample different graph topologies from our given distribution. Some translation circuits will exist in some of the sampled graphs, but not in others. This, in turn, means that a given vertex v will only be on a circuit for a fraction of the sampled graphs. We take the proportion of samples in which v is on a circuit to be the probability that v is in the translation set. We refer to this algorithm as *Unpruned* SenseUniformPaths (uSP).

3.2 Avoiding Correlated Sense-shifts

The second source of errors are circuits that include a pair of nodes sharing the same polysemy, *i.e.*, having the same pair of senses. A circuit might maintain sense s^* until it reaches a node that has both s^* and a distinct s_i. The next edge may lead to a node with s_i, but not s^*, causing an extraction error. The path later shifts back to sense s^* at a second node that *also* has s^* and s_i. An example for this is illustrated in Figure 2(e), where both the German and Swedish words mean feather and spring coil. Here, Italian 'penna' means only the feather and not the coil.

Two nodes that share the same two senses occur frequently in practice. For example, many languages use the same word for 'heart' (the organ) and center; similarly, it is common for languages to use the same word for 'silver', the metal and the color. These correlations stem from com-

[4] In our experiments we used a flat value of 0.6, chosen by parameter tuning on a development set of 50 inference tasks. In future we can use different values for different dictionaries based on our confidence in their accuracy.

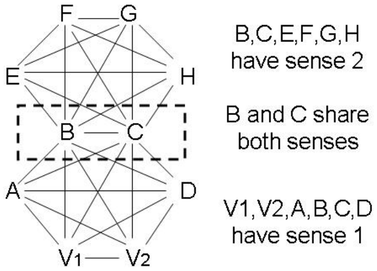

Figure 3: The set {B, C} has a shared ambiguity - each node has both sense 1 (from the lower clique) and sense 2 (from the upper clique). A circuit that contains two nodes from the same ambiguity set with an intervening node not in that set is likely to create translation errors.

mon metaphor and the shared evolutionary roots of some languages.

We are able to avoid circuits with this type of correlated sense-shift by automatically identifying *ambiguity sets*, sets of nodes known to share multiple senses. For instance, in Figure 2(e) 'Feder' and 'fjäder' form an ambiguity set (shown within dashed lines), as they both mean feather and coil.

Definition 2 *An* ambiguity set A *is a set of vertices that all share the same two senses. I.e.,* $\exists s_1, s_2$, with $s_1 \neq s_2$ s.t. $\forall v \in A$, *sense*$(v, s_1) \wedge$ *sense*(v, s_2)*, where* $sense(v, s)$ *denotes that* v *has sense* s.

To increase the precision of our algorithm we *prune* the circuits that contain two nodes in the same ambiguity set and also have one or more intervening nodes that are not in the ambiguity set. There is a strong likelihood that the intervening nodes will represent a translation error.

Ambiguity sets can be detected from the graph topology as follows. Each clique in the graph represents a set of vertices that share a common word sense. When two cliques intersect in two or more vertices, the intersecting vertices share the word sense of both cliques. This may either mean that both cliques represent the same word sense, or that the intersecting vertices form an ambiguity set. A large overlap between two cliques makes the former case more likely; a small overlap makes it more likely that we have found an ambiguity set.

Figure 3 illustrates one such computation. All nodes of the clique V_1, V_2, A, B, C, D share a word sense, and all nodes of the clique B, C, E, F, G, H also share a word sense. The set $\{B, C\}$ has nodes that have both senses, forming an ambiguity set. We denote the set of ambiguity sets by $\mathcal{A}$ in the pseudo-code.

Having identified these ambiguity sets, we modify our random walk scheme by keeping track of whether we are entering or leaving an ambiguity set. We prune away all paths that enter the same ambiguity set twice. We name the resulting algorithm SenseUniformPaths (SP), summarized at a high level in Algorithm 1.

Comparing Inference Algorithms Our evaluation demonstrated that SP outperforms uSP. Both these algorithms have significantly higher recall than TRANSGRAPH algorithm. The detailed results are presented in Section 4.2. We choose SP as our inference algorithm for all further research, in particular to create PANDICTIONARY.

3.3 Compiling PanDictionary

Our goal is to automatically compile PANDICTIONARY, a sense-distinguished lexical translation resource, where each entry is a distinct word sense. Associated with each word sense is a list of translations in multiple languages.

We use Wiktionary senses as the base senses for PANDICTIONARY. Recall that SP requires two nodes (v_1^* and v_2^*) for inference. We use the Wiktionary source word as v_1^* and automatically pick the second word from the set of Wiktionary translations of that sense by choosing a word that is well connected, and, which does not appear in other senses of v_1^* (*i.e.*, is expected to share only one sense with v_1^*).

We first run SenseUniformPaths to expand the approximately 50,000 senses in the English Wiktionary. We further expand any senses from the other Wiktionaries that are not yet covered by PANDICTIONARY, and add these to PANDICTIONARY. This results in the creation of the world's largest multilingual, sense-distinguished translation resource, PANDICTIONARY. It contains a little over 80,000 senses. Its construction takes about three weeks on a 3.4 GHz processor with a 2 GB memory.

Algorithm 1 S.P.$(G, v_1^*, v_2^*, \mathcal{A})$

```
parameters $N_G$: no. of graph samples, $N_R$: no. of random walks, $p_e$: prob. of sampling an edge
create $N_G$ versions of $G$ by sampling each edge independently with probability $p_e$
for all $i = 1..N_G$ do
    for all vertices $v$ : $rp[v][i] = 0$
    perform $N_R$ random walks starting at $v_1^*$ (or $v_2^*$) and pruning any walk that enters (or exits) an ambiguity set in $\mathcal{A}$ twice. All walks that connect to $v_2^*$ (or $v_1^*$) form a translation circuit.
    for all vertices $v$ do
        if($v$ is on a translation circuit) $rp[v][i] = 1$
return $\frac{\sum_i rp[v][i]}{N_G}$ as the prob. that $v$ is a translation
```

4 Empirical Evaluation

In our experiments we investigate three key questions: (1) which of the three algorithms (TG, uSP and SP) is superior for translation inference (Section 4.2)? (2) how does the coverage of PanDictionary compare with the largest existing multilingual dictionary, the English Wiktionary (Section 4.3)? (3) what is the benefit of inference over the mere aggregation of 631 dictionaries (Section 4.4)? Additionally, we evaluate the inference algorithm on two other dimensions – variation with the degree of polysemy of source word, and variation with original size of the seed translation set.

4.1 Experimental Methodology

Ideally, we would like to evaluate a random sample of the more than 1,000 languages represented in PanDictionary.[5] However, a high-quality evaluation of translation between two languages requires a person who is fluent in both languages. Such people are hard to find and may not even exist for many language pairs (*e.g.*, Basque and Maori). Thus, our evaluation was guided by our ability to recruit volunteer evaluators. Since we are based in an English speaking country we were able to recruit local volunteers who are fluent in a range of languages and language families, and who are also bilingual in English.[6]

The experiments in Sections 4.2 and 4.3 test whether translations in a PanDictionary have accurate word senses. We provided our evaluators with a random sample of translations into their native language. For each translation we showed the English source word and gloss of the intended sense. For example, a Dutch evaluator was shown the sense 'free (not imprisoned)' together with the Dutch word 'loslopende'. The instructions were to mark a word as correct if it could be used to express the intended sense in a sentence in their native language. For experiments in Section 4.4 we tested precision of *pairwise* translations, by having informants in several pairs of languages discuss whether the words in their respective languages can be used for the same sense.

We use the tags of correct or incorrect to compute the precision: the percentage of correct translations divided by correct plus incorrect translations. We then order the translations by probability and compute the precision at various probability thresholds.

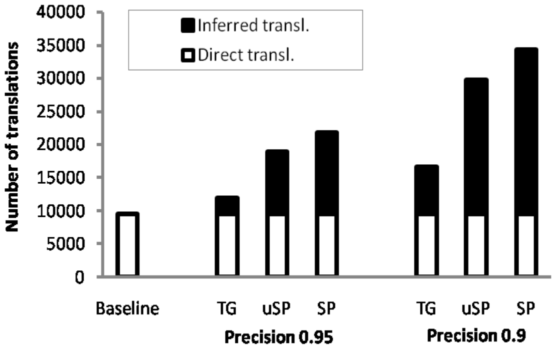

Figure 4: The SenseUniformPaths algorithm (SP) more than doubles the number of correct translations at precision 0.95, compared to a baseline of translations that can be found without inference.

4.2 Comparing Inference Algorithms

Our first evaluation compares our SenseUniformPaths (SP) algorithm (before and after pruning) with TransGraph on both precision and number of translations.

To carry out this comparison, we randomly sampled 1,000 senses from English Wiktionary and ran the three algorithms over them. We evaluated the results on 7 languages – Chinese, Danish, German, Hindi, Japanese, Russian, and Turkish. Each informant tagged 60 random translations inferred by each algorithm, which resulted in 360-400 tags per algorithm[7]. The precision over these was taken as a surrogate for the precision across all the senses.

We compare the number of translations for each algorithm at comparable precisions. The baseline is the set of translations (for these 1000 senses) found in the source dictionaries without inference, which has a precision 0.95 (as evaluated by our informants).[8]

Our results are shown in Figure 4. At this high precision, SP more than doubles the number of baseline translations, finding 5 times as many inferred translations (in black) as TG.

Indeed, both uSP and SP massively outperform TG. SP is consistently better than uSP, since it performs better for polysemous words, due to its pruning based on ambiguity sets. We conclude

[5] The distribution of words in PanDictionary is highly non-uniform ranging from 182,988 words in English to 6,154 words in Luxembourgish and 189 words in Tuvalu.

[6] The languages used was based on the availability of native speakers. This varied between the different experiments, which were conducted at different times.

[7] Some translations were marked as "Don't know".

[8] Our informants tended to underestimate precision, often marking correct translations in minor senses of a word as incorrect.

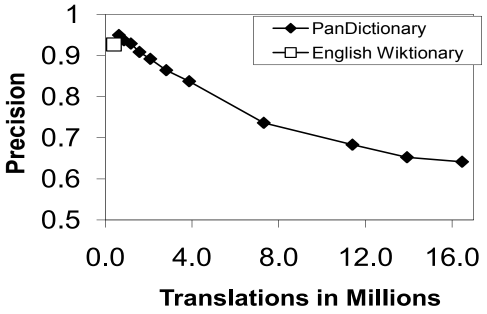

Figure 5: Precision *vs.* coverage curve for PANDICTIONARY. It quadruples the size of the English Wiktionary at precision 0.90, is more than 8 times larger at precision 0.85 and is almost 24 times the size at precision 0.7.

that SP is the best inference algorithm and employ it for PANDICTIONARY construction.

4.3 Comparison with English Wiktionary

We now compare the coverage of PANDICTIONARY with the English Wiktionary at varying levels of precision. The English Wiktionary is the largest Wiktionary with a total of 403,413 translations. It is also more reliable than some other Wiktionaries in making word sense distinctions. In this study we use only the subset of PANDICTIONARY that was computed starting from the English Wiktionary senses. Thus, this subsection under-reports PANDICTIONARY's coverage.

To evaluate a huge resource such as PANDICTIONARY we recruited native speakers of 14 languages – Arabic, Bulgarian, Danish, Dutch, German, Hebrew, Hindi, Indonesian, Japanese, Korean, Spanish, Turkish, Urdu, and Vietnamese. We randomly sampled 200 translations per language, which resulted in about 2,500 tags. Figure 5 shows the total number of translations in PANDICTIONARY in senses from the English Wiktionary. At precision 0.90, PANDICTIONARY has 1.8 million translations, 4.5 times as many as the English Wiktionary.

We also compare the coverage of PANDICTIONARY with that of the English Wiktionary in terms of languages covered. Table 1 reports, for each resource, the number of languages that have a minimum number of distinct words in the resource. PANDICTIONARY has 1.4 times as many languages with at least 1,000 translations at precision 0.90 and more than twice at precision 0.7. These observations reaffirm our faith in the panlingual nature of the resource.

PANDICTIONARY's ability to expand the lists of translations provided by the English Wiktionary is most pronounced for senses with a small number of translations. For example, at precision 0.90, senses that originally had 3 to 6 translations are increased 5.3 times in size. The increase is 2.2 times when the original sense size is greater than 20.

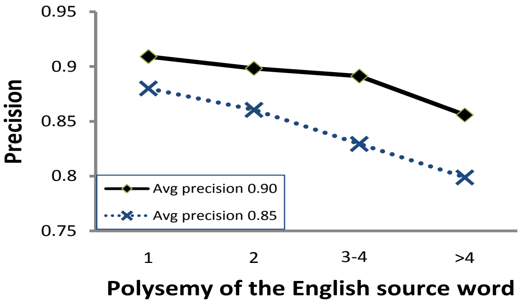

Figure 6: Variation of precision with the degree of polysemy of the source English word. The precision decreases as polysemy increases, still maintaining reasonably high values.

For closer analysis we divided the English source words (v_1^*) into different bins based on the number of senses that English Wiktionary lists for them. Figure 6 plots the variation of precision with this degree of polysemy. We find that translation quality decreases as degree of polysemy increases, but this decline is gradual, which suggests that SP algorithm is able to hold its ground well in difficult inference tasks.

4.4 Comparison with All Source Dictionaries

We have shown that PANDICTIONARY has much broader coverage than the English Wiktionary, but how much of this increase is due to the inference algorithm versus the mere aggregation of hundreds of translation dictionaries in PANDICTIONARY?

Since most bilingual dictionaries are not sense-distinguished, we ignore the word senses and count the number of distinct (word1, word2) translation pairs.

We evaluated the precision of word-word translations by a *collaborative tagging* scheme, with two native speakers of different languages, who are both bi-lingual in English. For each suggested translation they discussed the various senses of words in their respective languages and tag a translation correct if they found some sense that is shared by both words. For this study we tagged 7 language pairs: Hindi-Hebrew,

	# languages with distinct words		
	$\geq$ 1000	$\geq$ 100	$\geq$ 1
English Wiktionary	49	107	505
PanDictionary (0.90)	67	146	608
PanDictionary (0.85)	75	175	794
PanDictionary (0.70)	107	607	1066

Table 1: PANDICTIONARY covers substantially more languages than the English Wiktionary.

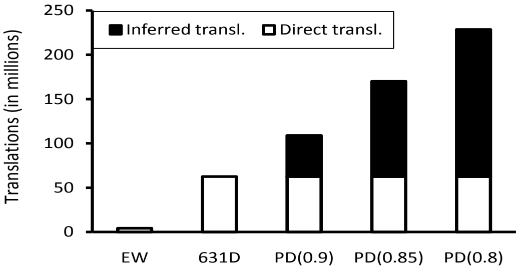

Figure 7: The number of distinct word-word translation pairs from PANDICTIONARY is several times higher than the number of translation pairs in the English Wiktionary (EW) or in all 631 source dictionaries combined (631 D). A majority of PANDICTIONARY translations are inferred by combining entries from multiple dictionaries.

Japanese-Russian, Chinese-Turkish, Japanese-German, Chinese-Russian, Bengali-German, and Hindi-Turkish.

Figure 7 compares the number of word-word translation pairs in the English Wiktionary (EW), in all 631 source dictionaries (631 D), and in PANDICTIONARY at precisions 0.90, 0.85, and 0.80. PANDICTIONARY increases the number of word-word translations by 73% over the source dictionary translations at precision 0.90 and increases it by 2.7 times at precision 0.85. PANDICTIONARY also adds value by identifying the word sense of the translation, which is not given in most of the source dictionaries.

5 Related Work

Because we are considering a relatively new problem (automatically building a panlingual translation resource) there is little work that is directly related to our own. The closest research is our previous work on TRANSGRAPH algorithm (Etzioni et al., 2007). Our current algorithm outperforms the previous state of the art by 3.5 times at precision 0.9 (see Figure 4). Moreover, we compile this in a dictionary format, thus considerably reducing the response time compared to TRANSGRAPH, which performed inference at query time.

There has been considerable research on methods to acquire translation lexicons from either MRDs (Neff and McCord, 1990; Helmreich et al., 1993; Copestake et al., 1994) or from parallel text (Gale and Church, 1991; Fung, 1995; Melamed, 1997; Franz et al., 2001), but this has generally been limited to a small number of languages. Manually engineered dictionaries such as EuroWordNet (Vossen, 1998) are also limited to a relatively small set of languages. There is some recent work on compiling dictionaries from monolingual corpora, which may scale to several language pairs in future (Haghighi et al., 2008).

Little work has been done in combining multiple dictionaries in a way that maintains word senses across dictionaries. Gollins and Sanderson (2001) explored using triangulation between alternate pivot languages in cross-lingual information retrieval. Their triangulation essentially mixes together circuits for all word senses, hence, is unable to achieve high precision.

Dyvik's "semantic mirrors" uses translation paths to tease apart distinct word senses from inputs that are not sense-distinguished (Dyvik, 2004). However, its expensive processing and reliance on parallel corpora would not scale to large numbers of languages. Earlier (Knight and Luk, 1994) discovered senses of Spanish words by matching several English translations to a WordNet synset. This approach applies only to specific kinds of bilingual dictionaries, and also requires a taxonomy of synsets in the target language.

Random walks, graph sampling and Monte Carlo simulations are popular in literature, though, to our knowledge, none have applied these to our specific problems (Henzinger et al., 1999; Andrieu et al., 2003; Karger, 1999).

6 Conclusions

We have described the automatic construction of a unique multilingual translation resource, called PANDICTIONARY, by performing probabilistic inference over the translation graph. Overall, the construction process consists of large scale information extraction over the Web (parsing dictionaries), combining it into a single resource (a translation graph), and then performing automated reasoning over the graph (SenseUniformPaths) to yield a much more extensive and useful knowledge base.

We have shown that PANDICTIONARY has more coverage than any other existing bilingual or multilingual dictionary. Even at the high precision of 0.90, PANDICTIONARY more than quadruples the size of the English Wiktionary, the largest available multilingual resource today.

We plan to make PANDICTIONARY available to the research community, and also to the Wiktionary community in an effort to bolster their efforts. PANDICTIONARY entries can suggest new translations for volunteers to add to Wiktionary entries, particularly if combined with an intelligent editing tool (*e.g.*, (Hoffmann et al., 2009)).

Acknowledgments

This research was supported by a gift from the Utilika Foundation to the Turing Center at University of Washington. We acknowledge Paul Beame, Nilesh Dalvi, Pedro Domingos, Rohit Khandekar, Daniel Lowd, Parag, Jonathan Pool, Hoifung Poon, Vibhor Rastogi, Gyanit Singh for fruitful discussions and insightful comments on the research. We thank the language experts who donated their time and language expertise to evaluate our systems. We also thank the anynomous reviewers of the previous drafts of this paper for their valuable suggestions in improving the evaluation and presentation.

References

E. Adar, M. Skinner, and D. Weld. 2009. Information arbitrage in multi-lingual Wikipedia. In *Procs. of Web Search and Data Mining(WSDM 2009)*.

C. Andrieu, N. De Freitas, A. Doucet, and M. Jordan. 2003. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50:5–43.

F. Bond, S. Oepen, M. Siegel, A. Copestake, and D D. Flickinger. 2005. Open source machine translation with DELPH-IN. In *Open-Source Machine Translation Workshop at MT Summit X*.

J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, and J. Frey. 2006. Context-based machine translation. In *AMTA*.

A. Copestake, T. Briscoe, P. Vossen, A. Ageno, I. Castellon, F. Ribas, G. Rigau, H. Rodriquez, and A. Samiotou. 1994. Acquisition of lexical translation relations from MRDs. *Machine Translation*, 3(3–4):183–219.

H. Dyvik. 2004. Translation as semantic mirrors: from parallel corpus to WordNet. *Language and Computers*, 49(1):311–326.

O. Etzioni, K. Reiter, S. Soderland, and M. Sammer. 2007. Lexical translation with application to image search on the Web. In *Machine Translation Summit XI*.

M. Franz, S. McCarly, and W. Zhu. 2001. English-Chinese information retrieval at IBM. In *Proceedings of TREC 2001*.

P. Fung. 1995. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *Proceedings of ACL-1995*.

W. Gale and K.W. Church. 1991. A Program for Aligning Sentences in Bilingual Corpora. In *Proceedings of ACL-1991*.

T. Gollins and M. Sanderson. 2001. Improving cross language retrieval with triangulated translation. In *SIGIR*.

Raymond G. Gordon, Jr., editor. 2005. *Ethnologue: Languages of the World (Fifteenth Edition)*. SIL International.

A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *ACL*.

S. Helmreich, L. Guthrie, and Y. Wilks. 1993. The use of machine readable dictionaries in the Pangloss project. In *AAAI Spring Symposium on Building Lexicons for Machine Translation*.

Monika R. Henzinger, Allan Heydon, Michael Mitzenmacher, and Marc Najork. 1999. Measuring index quality using random walks on the web. In *WWW*.

R. Hoffmann, S. Amershi, K. Patel, F. Wu, J. Fogarty, and D. S. Weld. 2009. Amplifying community content creation with mixed-initiative information extraction. In *ACM SIGCHI (CHI2009)*.

D. R. Karger. 1999. A randomized fully polynomial approximation scheme for the all-terminal network reliability problem. *SIAM Journal of Computation*, 29(2):492–514.

K. Knight and S. Luk. 1994. Building a large-scale knowledge base for machine translation. In *AAAI*.

I.D. Melamed. 1997. A Word-to-Word Model of Translational Equivalence. In *Proceedings of ACL-1997 and EACL-1997*, pages 490–497.

M. Neff and M. McCord. 1990. Acquiring lexical data from machine-readable dictionary resources for machine translation. In *3rd Intl Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*.

P. Vossen, editor. 1998. *EuroWordNet: A multilingual database with lexical semantic networds*. Kluwer Academic Publishers.