Panlingual lexical translation via probabilistic inference

Mausam [*], Stephen Soderland, Oren Etzioni, Daniel S. Weld, Kobi Reiter [1], Michael Skinner [1], Marcus Sammer, Jeff Bilmes

Department of Computer Science and Engineering, Box 352350, University of Washington, Seattle, WA 98195, United States

A B S T R A C T

This paper introduces a novel approach to the task of lexical translation between languages for which no translation dictionaries are available. We build a massive *translation graph*, automatically constructed from over 630 machine-readable dictionaries and Wiktionaries. In this graph each node denotes a word in some language and each edge (v_i, v_j) denotes a word sense shared by v_i and v_j. Our current graph contains over 10,000,000 nodes and expresses more than 60,000,000 pairwise translations.

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 describes a series of probabilistic inference algorithms that solve this problem at varying precision and recall levels. All algorithms enable us to quantify our confidence in a translation derived from the graph, and thus trade precision for recall.

We compile the results of our best inference algorithm to yield PANDICTIONARY, a novel multilingual dictionary. PANDICTIONARY contains more than four times as many translations as 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.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In the era of globalization, inter-lingual communication is becoming increasingly important. Nearly 7000 languages are in use today [18] necessitating machine translation (MT) systems between about 49 million language-pairs. In contrast popular MT systems like Google Translate handle only on the order of a thousand language pairs. It is difficult to see how statistical Machine Translation (MT) methods can scale to this large number of language pairs, since they depend on aligned corpora, which are very expensive to generate, and are available at the requisite scale for only a tiny number of language pairs [5,28,33,30,7].

This paper considers scaling MT in the context of a far easier task: *lexical translation*. Lexical translation is the task of translating individual words or phrases (*e.g.*, "sweet potato") from one language to another. Because lexical translation does not require aligned corpora as input, it is feasible for a much broader set of languages than statistical MT. While lexical translation has a long history (*cf.* [24,20,9,23]), interest in it peaked in the 1990s. Yet, as this paper shows, the proliferation of Machine-Readable Dictionaries (MRDs) and the rapid growth of multilingual Wiktionaries offers the opportunity to scale lexical translation to an unprecedented number of languages.

---

* Corresponding author. Tel.: +1 206 685 1964; fax: +1 206 543 2969.
  E-mail address: mausam@cs.washington.edu (Mausam).
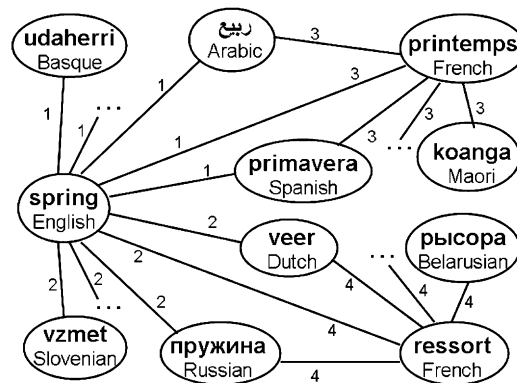[1] Current address: Google Inc., 651 N 34th St., Seattle, WA 98105, USA.

Fig. 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.

Of course, lexical translation cannot replace statistical MT, but it is useful for several applications, including the translation of search-engine queries, meta-data tags,[2] library classifications and recent applications like cross-lingual image search [11] of *http://www.panimages.org*, and enhancement of multilingual Wikipedias [1]. Also, lexical translation is a valuable component in knowledge-based Machine Translation (MT) systems, *e.g.*, [4,6]. The increasing international adoption of the Web yields opportunities for new applications of lexical translation systems.

The fundamental contribution of this paper is a novel approach to lexical translation, which automatically compiles various machine-readable multilingual and bilingual dictionaries available on the Web into a unique *translation graph*. A node v in the translation graph represents a word in a particular language. An edge (v_i, v_j) denotes a word sense shared by v_i and v_j. Fig. 1 shows a snippet of the translation graph. We demonstrate that inference over this translation graph can yield a massive, multilingual dictionary with coverage superior to the union of input dictionaries at comparable precision.

Inference over the translation graph necessitates matching word senses across multiple, independently-authored dictionaries. For example, if one dictionary says that 'udaherri' and 'printemps' translate 'spring' another says that 'koanga' and 'spring' are translations of 'printemps', then we need to infer whether the two dictionaries are referring to the same sense – resulting in 'udaherri' a translation of 'koanga', or not (see Fig. 1). 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 key technical contribution of this paper is a set of methods that perform probabilistic sense matching to *infer* lexical translations between two languages that do not share a translation dictionary. For example, our algorithm can conclude that the Basque word 'udaherri' is a translation of the Maori word 'koanga'.

We present three different techniques for probabilistic inference – TRANSGRAPH, unpruned SENSEUNIFORMPATHS (uSENSEUNIFORMPATHS) and SENSEUNIFORMPATHS. TRANSGRAPH uses heuristic-based formulae for inference, while the second, uSENSEUNIFORMPATHS, reasons about graph topology via random walks and probabilistic graph sampling. SENSEUNIFORM-PATHS adds constraints based on the graph topology on uSENSEUNIFORMPATHS that improve precision.

We use SENSEUNIFORMPATHS to construct 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.

This paper combines and extends our previous two papers [11,31] and overall, makes the following contributions:

1. We introduce a novel approach to the task of lexical translation, which compiles a large number of machine readable dictionaries in a single resource called a translation graph. We employ probabilistic reasoning and inference over the translation graph to infer translations that are not expressed by any of the input dictionaries.
2. We develop three inference algorithms: TRANSGRAPH, unpruned SENSEUNIFORMPATHS, and SENSEUNIFORMPATHS. All these algorithms return new translations with associated confidence values, so we can trade precision for recall. We empirically compare the three algorithms and find that SENSEUNIFORMPATHS outperforms the others by returning many more translations at high precisions.
3. We use SENSEUNIFORMPATHS to compile PANDICTIONARY – a massive, sense-distinguished multilingual dictionary. Our empirical evaluations show that depending on the desired precision PANDICTIONARY is 4.5 to 24 times larger than the English Wiktionary (http://en.wiktionary.org). Moreover, it expresses about 4 times the number of pairwise translations compared to the union of its input dictionaries (at precision 0.8).

The remainder of the paper is organized as follows. Section 2 introduces the construction of the translation graph. We describe the three methods for inference and compare them in Section 3. Section 4 describes the compilation of PANDIC-

---

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

TIONARY and compares its coverage with the English Wiktionary. Section 5 considers related work on lexical translation. The paper concludes in Sections 7 and 6 with conclusions and directions for future work.

2. The translation graph

This section describes the properties of translation graph and its construction from multiple dictionaries. The translation graph is an undirected graph defined as a triple $\langle \mathcal{V}, \mathcal{E}, \Psi \rangle$.

$\mathcal{V}$ and $\mathcal{E}$ denote the usual sets of vertices and edges. 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. Additionally, an edge is labeled by an integer denoting an ID for the word sense. Ψ is a set of *inequality constraints* between sense IDs. It is a set of pairs of sense IDs, such that if the pair $\langle id_1, id_2 \rangle \in \Psi$ then the senses represented by the IDs are known to be distinct, *i.e.*, they represent different word senses.

Fig. 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. For this fragment $\Psi = \{\langle 1, 2 \rangle, \langle 3, 4 \rangle\}$.

Note that sense-distinguished multilingual entries give rise to cliques all of which share a common sense ID. In Fig. 1 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'. This graph grows rapidly; for instance, the English Wiktionary entry for the season sense of 'spring' has 58 translations and thus 1653 (58 choose 2) edges.

We build the translation graph incrementally on the basis of entries from multiple, independent dictionaries (as described below). As edges are added on the basis of entries from a new dictionary, some of the new word sense IDs are redundant because they are equivalent to word senses already in the graph from another dictionary. This leads to the following semantics for sense IDs: if two edges have the same ID then they represent the same sense, however, if two edges have different IDs, they may or may not represent the same sense (except if they belong to Ψ, in which case they represent distinct senses). For example, our translation graph assigns one word sense ID to the seasonal sense of 'spring' from an English dictionary, a new word sense ID to the French dictionary entry for 'printemps', and so forth (see labels '1' and '3' in Fig. 1). We refer to this phenomenon as *sense ID inflation*.

2.1. Construction of the translation graph

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 (http://wiktionary.org) are multilingual dictionaries created by volunteers collaborating over the Web, which provide translations from a source language into multiple target languages, generally distinguishing between different word senses. 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.

As each new sense-distinguished dictionary is added to the graph we assign it a new, unique word sense ID for each word sense from that dictionary. Thus, edges for translations of the season 'spring' from the English Wiktionary have one word sense ID, edges for translations of the flexible coil 'spring' have a different word sense ID, and so forth. If two entries come from the same dictionary and have the same source word, they likely represent multiple word-senses for the source node. In such a case we add inequality constraints in Ψ for all pairs of such entries. For the dictionaries that are not word-sense distinguished, *e.g.*, in the case of most bilingual dictionaries, we assign a new sense ID for each translation (increasing sense ID inflation).

Some multilingual dictionaries fail to separate the different senses of a word. For example, the French Wiktionary has an entry for the word 'boule' with English translations as 'ball', 'bowl', 'chunk', 'clod', and 'lump'. These are all good translations of 'boule', but clearly not all in the same sense. We use a simple heuristic to detect these "impure" entries: a multilingual entry is considered impure if it has more than three translations in the same language. In such cases we create a new ID for each edge, essentially treating the translations as if they came from a bilingual dictionary, and moreover, we do not add edges between the various translations – thus forming a spider instead of a clique.

As pointed out earlier, sense ID inflation poses a challenge for inference in translation graphs. If we wish to find all words that are translations of the sense (say s^*) represented by a given sense ID we need to look for sources of evidence that help us determine that another sense ID also represents s^*. We develop three algorithms for this inference task that we describe in Section 3.

3. The inference algorithms

Our inference task is defined as follows: given a sense ID, say id^*, that represents a sense, say s^*, compute the translations (in different languages) of s^*. We describe three algorithms for inference over the translation graph. Section 3.1
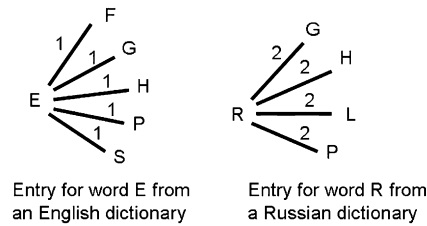
Fig. 2. Schematic diagram of edges from an entry for the word *E* from an English dictionary and edges from an entry for the word *R* from a Russian dictionary.
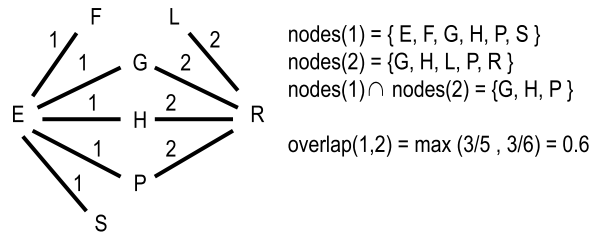

Fig. 3. After the entries from Fig. 2 have both been added to the graph, the set of nodes with word sense ID 1 overlaps with the set of nodes for word sense ID 2. The proportion of overlapping nodes gives evidence that the two word senses may be equivalent.

describes TRANSGRAPH, which is based on formulae for sense ID equivalence, *i.e.*, scores whether a pair of sense IDs from the translation graph refers to the same sense. Our other two algorithms, uSENSEUNIFORMPATHS and SENSEUNIFORMPATHS (Sections 3.5 and 3.6), are based on a graph sampling and random walk scheme that is based on a theory of translation circuits. We motivate our theoretical formulation by a set of representative examples in Section 3.2 and describe the theoretical results in Section 3.3. Finally, Section 3.7 presents our results comparing the three algorithms at different precision levels.

3.1. TRANSGRAPH

Recall that each vertex is associated with several sense IDs (for all the edges that are incident on it). In the TRANSGRAPH method we compute sense ID equivalence scores of the form $score(id_i \equiv id_j)$. If a vertex has a sense ID that is same as id^* with a high score then it is a likely translation of s^*.

Figs. 2 and 3 give a schematic illustration of how TRANSGRAPH accumulates entries from multiple dictionaries. Fig. 2 shows graph edges from a multilingual entry for the word *E* from an English dictionary that gives translations into French, German, Hungarian, Polish, and Spanish. TRANSGRAPH assigns the word sense ID 1 for these edges. This figure also shows edges from an entry for word *R* from a Russian dictionary, which in this case has translations into German, Hungarian, Latvian, and Polish. These edges are assigned word sense ID 2.

Fig. 3 shows the situation after both sets of edges have been added to the translation graph. There are 6 nodes with edges labeled with word sense ID 1, $\{E, F, G, H, P, S\}$; 5 nodes with edges labeled 2, $\{G, H, L, P, R\}$; and an intersection of these sets comprising 3 nodes, $\{G, H, P\}$. The three nodes in the intersection have two incident edges with distinct sense IDs 1 and 2. The proportion of intersecting nodes provides evidence that these IDs refer to the same word sense.

TRANSGRAPH estimates whether two multilingual word sense IDs id_i and id_j are equivalent by assigning an equivalence score between 0 and 1 as follows:

- A word sense is equivalent to itself: $score(id \equiv id) = 1$.
- If id_i and id_j are alternate word senses from the same entry in a sense-distinguished dictionary (*i.e.*, $\langle i, j \rangle \in \Psi$), then they are distinct: $score(id_i \equiv id_j) = 0$.
- If word senses id_i and id_j have at least K intersecting nodes, then set the score by Eq. (1) below.
- In all other cases, the score is undefined.

TRANSGRAPH estimates the score that id_i and id_j represent the same word senses by the following equation. If $|nodes(id_i) \cap nodes(id_j)| \geqslant K$, then:

$$score(id_i \equiv id_j) = \max\left(\frac{|nodes(id_i) \cap nodes(id_j)|}{|nodes(id_i)|}, \frac{|nodes(id_i) \cap nodes(id_j)|}{|nodes(id_j)|} \right) \tag{1}$$

where $nodes(id)$ is the set of nodes that have edges labeled by word sense ID id, and K is a sense intersection threshold. In our experiments K was chosen after examining a small sample of the translation graph.

As an example of computing the score of sense ID equivalence, our translation graph has 56 translations for the season sense of 'spring' from an English dictionary, and 12 translations for 'printemps' from a French dictionary. Eight of these translations overlap, giving a score of $\frac{8}{12} = 0.67$ that the two senses are equivalent.

3.1.1. Computing translation scores

Given the translation graph coupled with the sense ID equivalence scores, TRANSGRAPH can now score a word as a translation of another word in a given word sense. First, we show how to compute the translation score of a single translation path. Then, we show how we combine evidence across multiple paths.

We utilize the following observations in reasoning about paths in the graph:

- If word A is translated as B, and B is translated to a third language as C, it follows that A can be translated as C if the word sense has not changed when translating from A to B and from B to C. Otherwise, A may translate to one sense of B, but another sense of B translates to C. In this case, A and C may have entirely different meanings. Fig. 1 has examples of this: 'primavera' is translated as 'spring' in the season sense, while 'spring' is translated as 'ressort' in the flexible coil sense, but 'primavera' and 'ressort' are not translations of each other.
- Translation dictionaries have limited coverage – the lack of a translation between word A and word B cannot be taken as evidence that A is not translated as B.

Consider a single path P that connects vertex v_1 to v_k, where v_i is the word w_i in language l_i and the ith edge has sense ID id_i. Let $pathScore(v_1, v_k, id^*, P)$ be the score of (w_1, l_1) as a correct translation of (w_k, l_k) in word sense represented by sense ID id^*, given a path P connecting these nodes.

The simple case is where the path is of length 1. If id^* is the same sense ID as id_1, then the score is simply 1.0; otherwise it is the score of equivalence of the two senses are equivalent:

$$pathScore(v_1, v_2, id^*, P) = score(id^* \equiv id_1), \qquad (2)$$

where the path P has more than one edge, the path score is reduced by $score(id_i \equiv id_{i+1})$ whenever the word sense ID changes along the path. We make the simplifying assumption that sense-equivalence scores are mutually independent. Formally, this gives the term

$$\prod_{i=1\ldots|P|-1} score(id_i \equiv id_{i+1}).$$

If the desired sense s^* (represented by id^*) is not found on the path, we also need to factor in a score that id^* is equivalent to at least one sense ID id_i on the path, which we approximate by the maximum of $score(id^* \equiv id_i)$ over all id_i. Formally, this gives the term

$$\max_{i=1\ldots|P|} \left(score(id^* \equiv id_i) \right),$$

which is equal to 1.0 if id^* is found on path P.

Putting these two terms together, we have the following formula for simple paths of length greater than one (*i.e.*, $|P| > 1$):

$$pathScore(v_1, v_k, id^*, P) = \max_{i=1\ldots|P|} \left(score(id^* \equiv id_i) \right) \times \prod_{i=1\ldots|P|-1} score(id_i \equiv id_{i+1}). \qquad (3)$$

Note that we disallow paths that contain non-consecutive repetition of sense IDs (*e.g.*, 1, 2, 1).

There are typically multiple paths from one node to another in the translation graph. The simplest way to compute $score(v_1, v_k, id^*)$ is to take the maximum score of any path between id_1 and id_k,

$$score(v_1, v_k, id^*) = \max_{P \in paths} \left(pathScore(v_1, v_k, id^*, P) \right). \qquad (4)$$

We experimented with another method that gives a higher score if there are multiple, *distinct* paths between words. We define two paths from v_1 to v_k to be distinct if there is a distinct sequence of unique word sense IDs on each path. We combined scores using a standard Noisy–Or model. The basic intuition is that translation is correct unless every one of the translation paths fails to maintain the desired sense s^*. We multiply the score of failure $(1 - pathScore)$ for each path. We then subtract that score from one to get a new score for the correct translation. The translation score of v_1 as a translation of v_k in word sense s^* is:

$$score(v_1, v_k, id^*) = 1 - \prod_{P \in distinct\, P} \left(1 - pathScore(v_1, v_k, id^*, P) \right), \qquad (5)$$

where *distinct P* is the set of distinct paths from v_1 to v_k.
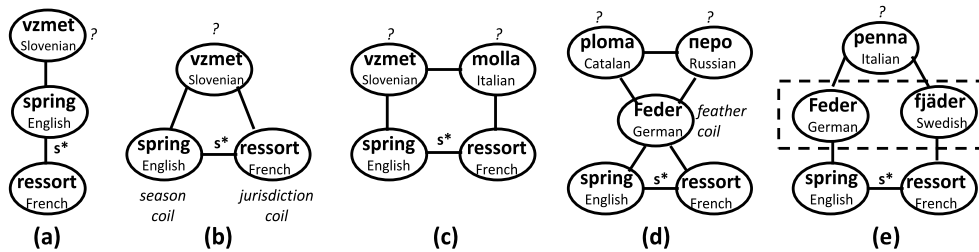
Fig. 4. Snippets of translation graphs illustrating various inference scenarios. The nodes in question mark represent the nodes in focus for each illustration. For all cases we are trying to infer translations of the flexible coil sense of spring.

We found that our current implementation of the Noisy–Or model tends to give inflated scores, so we use the maximum path score in the experiments reported in the paper. Defining distinct paths as those with distinct sense IDs is not sufficient to ensure that paths are based on independent evidence. We describe a better method to incorporate distinct evidence in the next section.

3.1.2. Bilingual dictionaries

The method for computing sense ID equivalence discussed above holds only for multilingual dictionaries, in which multiple translations per sense ID are present. Unfortunately, we do not always have this luxury. For bilingual dictionaries the sense IDs may only appear once in the translation graph. In response, we identify 3-cliques in the graph as an additional structure that helps to combat sense ID inflation.

Consider, for example, the simple clique shown in Fig. 4(b). The figure shows a 3-node clique where each of the edges was derived from a distinct dictionary, and hence has a distinct word sense ID. The edge from ('spring', English) to ('ressort', French) is labeled id^* (representing the sense s^*) and comes from an entry for the flexible coil of spring from the English Wiktionary. The edge between ('vzmet', Slovenian) to ('spring', English) is from a Slovenian–English dictionary that does not specify which sense of spring is intended. The third edge is from a Slovenian–French dictionary, again without any indication of word sense.

Based on this evidence only, the probability is high that 'vzmet' has the sense s^*. It has long been known that this kind of *triangulation* gives a high probability that all three words share a common word sense [17]. We revisit this example in more detail in the next section.

We empirically estimated the probability that all three word sense IDs of a 3-node clique are equivalent to be approximately 0.80 in our current translation graph. This number was computed by randomly sampling 3-node cliques in languages for which we had in-house experts and testing whether they preserved the initial sense. The TRANSGRAPH compiler finds all cliques in the graph of size 3 where two word senses are from bilingual dictionaries. It then adds an entry to the sense ID equivalence table with probability 0.80 for each pair of sense IDs in the clique. These probabilities are then used as equivalence scores in Section 3.1.1 to compute translation scores.

TRANSGRAPH is the first method that performs scalable inference for lexical translation. It is based on two formulas: one, which computes a score that two multilingual entries represent the same word sense, and two, which estimates the probability that three edges forming a triangle represent the same word sense. Preliminary experiments (see Section 3.7) showed that the algorithm was able to infer several translations, which were not asserted by any single dictionary, at high precision. However, in subsequent work, we identified several places for improvement in the algorithm:

- The translation scores from different paths are combined conservatively (either taking the max over all paths, or using "Noisy–Or" on paths that are completely disjoint). An ideal algorithm will combine evidence over both dependent and independent paths by handling the dependencies accurately.[3]
- The formulae of TRANSGRAPH 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.
- The insights behind the triangles will benefit from a theoretical formalization.
- As reported in Section 3.7, at high precision, TRANSGRAPH is able to infer a relatively small fraction of new translations.

We now use this critique as the guiding principles to, first, develop a set of theoretical insights about our translation problem. We formalize these notions based on an idealized semantics. Finally, we extend TRANSGRAPH into two novel algorithms – unpruned SENSEUNIFORMPATHS and SENSEUNIFORMPATHS, which achieve substantially higher recall at comparable high precisions.

---

[3] Two paths are independent if they do not share any edges.

3.2. Insights from translation graph snippets

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 Fig. 4(a) if A is the French word 'ressort' (means both jurisdiction and 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 (and no sense ID equivalence probabilities) 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 Fig. 4(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 Fig. 4(c), both 'molla' and 'vzmet' have the sense coil, even though no explicit triangle is present. The path from 'spring' through 'vzmet', 'molla', and 'ressort' completes a circuit in the graph and returns to 'spring'. We have the same two cases as we had for graph triangles: either all nodes share a common sense, or there is an unlikely combination of senses for nodes on the path that allow the circuit to complete. For example, 'vzmet' may mean both coil and jurisdiction and 'molla' means jurisdiction, but not coil. To formalize these intuitions, 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^*.

Our intuition from triangles and from paths of four nodes can be extended to translation circuits of arbitrary length. All nodes on a translation circuit share a sense with high probability, unless there is correlated polysemy among the nodes on the path. We begin by assuming that polysemy is completely uncorrelated, and on this basis we are able to develop a mathematical model of sense-assignment that lets us formally prove theorems based on these insights. Later in Section 3.6 we introduce a mechanism to detect and avoid correlated polysemy using graph topology and hence are able to relax our assumption.

3.3. Theory of translation inference

This section presents a formal model of the translation inference problem. This model captures our insights regarding translation circuits, but under two explicit simplifying assumptions. We note that the assumptions make this an idealized model, since they do not capture the actual process through which languages developed and share translations. However, in subsequent subsections we successively relax these assumptions to yield an inference algorithm with strong performance in practice.

At the highest level, we model the fact that each word in any language represents several senses. For this we consider a set of language-independent senses, $\mathcal{S}$. This set represents all the true senses a word can express. This will be a large set, since the number of concepts requiring expression is huge. Given this set and sense $s \in \mathcal{S}$ we denote $sense(v, s)$ to denote that the vertex v expresses sense s.

Our random variable assigns each word to a set of senses, such that this assignment is compatible with the observed translation graph. In other words if there is an edge between two vertices v_1 and v_2 in the graph then the sense assignment makes sure that there is at least one common sense between v_1 and v_2. We ask: given a translation graph and the knowledge that two vertices, v_1^* and v_2^*, share only the sense s^*, what is the probability that vertices in the translation circuit do not share s^*?

To answer this question we make two idealized assumptions. Our first assumption is that all edges in the translation graph indicate true translations.

Assumption 1 *(Edge correctness).* $(v_1, v_2) \in \mathcal{E} \Rightarrow \exists s$ s.t. $sense(v_1, s) \land sense(v_2, s)$.

This assumption is often true and is violated in situations where the quality of the dictionary is low, or the extractor scraping the dictionaries makes many errors in extracting translations. Our second assumption states that the polysemy of different words is not correlated.
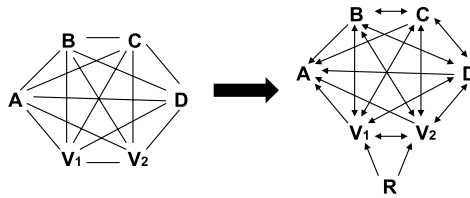
Fig. 5. Computing the existence of a translation circuit through *A* is converted into an equivalent flow problem on a directed graph with an additional node *R*.

Assumption 2 *(Polysemy uniformity).* If a vertex has total *m* senses then knowing *i* of them does not predict the rest of the senses it has. More formally, let *v* be a vertex with a set of known senses *S* ($|S| = i$) and let S_1 and S_2 denote sets of senses that are disjoint from *S*, s.t. $|S_1| = |S_2| = m - i$. Then $Pr(sense(v, S_1)|sense(v, S)) = Pr(sense(v, S_2)|sense(v, S))$

Polysemy uniformity captures a generative model in which the assignment of senses to words is done uniformly at random and independent of other words in other languages. We know that languages did not evolve independently or by random assignment – rather they co-developed by word sharing, word transformations and metaphorical usages. Hence, this theoretical model is idealized. Still, this idealization lets us prove the following theorem about translation circuits, which forms the basis of our next two algorithms. In Sections 3.5 and 3.6 we successively relax these two assumptions to develop algorithms that perform very well in practice.

Theorem 1. *Let C^k be a translation circuit of length k ($k \ll |\mathcal{S}|$) with origin v^* and sense s^*. Let P be the set of vertices along this circuit, let $|\mathcal{S}|$ denote the number of possible word senses for all words in all languages, and let the maximum number of senses per word be bounded by N ($N \ll |\mathcal{S}|$). Then $\forall v \in P \lim_{|\mathcal{S}| \to \infty} Pr(sense(v, s^*)) = 1$ (under Assumptions 1 and 2).*

Proof sketch. Any erroneous translation circuit C^k that begins with s^* includes a series of nodes that do not have s^*. Call the last of these nodes v_i. An edge (v_i, v_{i+1}) leads to node v_{i+1} that does have s^*.

By Assumption 1, (v_i, v_{i+1}) can exist only if v_{i+1} shares a sense with v_i, which we show is highly unlikely. If v_{i+1} has s^* and $n - 1$ other senses, there are $\frac{(|\mathcal{S}|-1)!}{(|\mathcal{S}|-n)!(n-1)!}$ combinations of senses, which are equally likely by Assumption 2. If v_{i+1} includes one of the senses from v_i, there are $\frac{(|\mathcal{S}|-2)!}{(|\mathcal{S}|-n)!(n-2)!}$ combinations for its remaining $n - 2$ senses. This gives a probability of $\frac{n-1}{|\mathcal{S}|-1}$ that a given sense of v_i matches, and a probability bounded by $\frac{m(n-1)}{|\mathcal{S}|-1}$ that v_{i+1} has one of the at most *m* senses of v_i. This probability tends to zero as $|\mathcal{S}| \to \infty$, so the probability of an error-free translation circuit tends to 1.0. □

We provide the complete proof in Appendix A.

3.4. The basic translation algorithm

These insights and the theorem suggest a basic version of our algorithm: "given two vertices, v_1^* and v_2^*, that share a single 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^*". This algorithm forms the basis for all further algorithmic extensions.

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 Fig. 4(d) the Catalan word 'ploma' has paths to both spring and ressort, but there is no translation circuit through it. Hence, it will not be considered a translation. This example also illustrates potential errors avoided by the basic 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 will enumerate all paths from v_1^* to v_2^* and would be too slow. We can, alternatively, convert the problem of testing the existence of a translation circuit as a flow problem. We create a new node *r*. We add outgoing edges from *r* to v_1^* and from *r* to v_2^*. We assign each node (except *r* and *v*, the vertex under consideration) with a unit capacity. All connections to *v* are incoming to *v* whereas all other edges are bidirectional. In this directed graph if we can send 2 units of flow from *r* to *v* then there exists a translation circuit between v_1^* and v_2^* that goes through *v*. Fig. 5 illustrates the flow formulation schematically.[4]

The best known complexity of solving a single flow problem with node capacities is $O(|\mathcal{E}|^{3/2} \log |\mathcal{E}|)$ [16]. Since we need to run this procedure once per vertex (*v*) overall the complexity of this procedure is $O(|\mathcal{V}||\mathcal{E}|^{3/2} \log |\mathcal{E}|)$. While polynomial this will still be too costly to run on graphs of our sizes. We approximate the solution by a random walk scheme. We start

---

[4] We thank Rohit Khandekar for suggesting this flow formulation.

a random walk from v_1^* (or v_2^*) and choose random edges (with uniform probability) 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.

The time complexity of this scheme is $O(kDN_R)$ where N_R is the number of random walks, D is the max degree of any vertex and k is the max length of the random walk. Notice that this complexity is independent of the size of the graph and depends only a local property, the degree of a vertex.

In our implementation we performed a total of 4000 random walks of max circuit length 7. We chose these parameters based on a development set of 50 inference tasks. Please refer to Section 3.8 for a control experiment varying these parameters.

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. *i.e.*, both of our assumptions are violated in real data. Below we add two new features to our algorithm to deal with each of these error sources, respectively.

3.5. *Unpruned* SENSEUNIFORMPATHS*: Errors in source dictionaries*

In practice, source dictionaries contain mistakes and errors occur in processing the dictionaries to create the translation graph. This is especially true for dictionaries automatically generated from parallel texts. Our algorithm is able to handle noise in dictionaries using the insight that existence of a *single* translation circuit is only limited evidence for a vertex as a translation. Instead, several translation circuits constitutes a 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. 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 (edge noise) represents multiple noisy graph transformations, *i.e.*, 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.

Computing probabilities of graph properties over random graphs has been studied in the literature (*e.g.*, [3]), but we have not found a published method to tractably compute the probability of existence of a translation circuit through a vertex. Thus, we use a graph sampling approach and our random walk scheme within each sample to estimate this probability for each vertex.

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).

Algorithm 1 describes the sampling scheme in which each edge is sampled with some probability and Algorithm 2 describes the probability computation using sampling as a subroutine. In assigning probabilities of sampling edges, we make a distinction between *cliques* that were constructed based on an entry in a multilingual dictionary and *single edges* that were constructed based on an entry from a bilingual dictionary. Since the edges in a clique are not based on independent evidence, we sample edges in a clique differently than single edges, as detailed in Algorithm 1.

In our implementation we used a flat value of 0.6 for both p_c and p_s in Algorithm 1. This value was 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.

Algorithm 1. Sample Graphs(G, N_G).

1: **for all** $i = 1..N_G$ **do**
2:    **for all** single edges $e \in G$ **do**
3:       add e to G_i with probability p_s.
4:    **for all** multilingual cliques $c \in G$ **do**
5:       **for all** vertices $v \in c$ **do**
6:          sample $present_c(v)$ with probability p_c
7:       **for all** pairs of vertices $v_1, v_2 \in c$ **do**
8:          add $e(v_1, v_2)$ to G_i if both $present_c(v_1) = 1$ and $present_c(v_2) = 1$
9: **return** $\{G_i\}_{i=1}^P$ as the N_G sampled graphs

Recall that we compute the existence of simple circuits by a random walk scheme. Line 5 of Algorithm 2 suggests that we need to execute the random walk algorithm for each graph sample. In fact, we can optimize this further. Given enough memory we can get away with performing the set of random walks only once (on the original graph G). For every translation circuit found in G we test if all the edges are present in each sample G_i. If they are then we set the $rp[v][i]$ bit to 'true'. Counting the number of true bits for each vertex will give us the numerator for the probability (line 7 in Algorithm 2).

Algorithm 2. Unpruned SenseUniformPaths(G, v_1^*, v_2^*, N_G).

1: *parameters* N_G: no. of graph samples, N_R: no. of random walks, p_s, p_c: prob. of sampling an edge and node for a single edge and cluster respectively.
2: **for all** $v \in V$, $rp[v] = 0$
3: Sample Graphs(G, N_G)
4: **for all** $i = 1..N_G$ **do**
5:    perform N_R random walks starting at v_1^* (or v_2^*). All walks that connect to v_2^* (or v_1^*) form a translation circuit.
6:    **for all** vertices v, if v is on a translation circuit $(v_1^*, v_2^*) \in G_i$, $rp[v][i] + +$
7: **return** $\frac{\sum_i rp[v][i]}{N_G}$ as the probability that v is a translation

Another optimization avoids storing all graph samples in memory. Instead, it stores all translation circuits in G and samples only the subgraph that is active in at least 1 circuit. This saves significant memory and also graph sampling time.

This algorithm can be divided into four parts – sampling, random walk, setting bits after each random walk, and lastly, computing the probability. Sampling time is $O(N_G|\mathcal{E}|)$, random walk takes $O(kDN_R)$, setting of bits requires $O(kN_R N_G)$, and computing the probability takes $O(N_G|\mathcal{V}|)$.

Overall, the time complexity is (pseudo-)linear in the size of the graph and hence runs quite fast in practice. Moreover, we can easily trade running time with quality of approximation by varying the number of random walks and graph samples.

3.6. SenseUniformPaths: *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 Fig. 4(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 common 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 Fig. 4(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.

Fig. 6 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.

Note that this method is able to identify only a subset of ambiguity sets, since if two sense IDs represent same sense but share few words in common then we will miss those ambiguity sets. However, in practice this scheme is able to identify many such sets and give a boost to the quality of results.

As a source of additional evidence for avoiding sense-shift, we make use of Ψ – the set of sense-pairs that are asserted to be distinct by a dictionary. These are cliques that were constructed from alternate senses for a word from a multilingual dictionary. We prune a random walk if it visits an edge from sense ID id_1 and has already visited an edge from id_2 and if $\langle id_1, id_2 \rangle \in \Psi$.

Our preliminary experiments revealed that both prunings – pruning a walk if it enters an ambiguity set twice, and pruning a walk if it hops between sense IDs that are known to be distinct – are effective in making fewer errors. The combination of the two is the most effective. We name the resulting algorithm SenseUniformPaths.

Implementation. In contrast to TransGraph, both uSenseUniformPaths and SenseUniformPaths require two special vertices v_1^* and v_2^* from the input sense ID (id^*) for inference. We require the two vertices to have the following desirable properties:
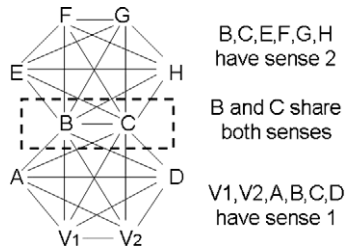
Fig. 6. The set $\{B, C\}$ has a shared ambiguity – each node has both sense 1 (from the lower clique) and sense 2 (from the upper clique). A circuit that contains two nodes from the same ambiguity set with an intervening node not in that set is likely to create translation errors.

- They have only one sense in common, otherwise the inference will end up mixing the word senses these two words share.
- They are well-connected to other vertices in the graph, or else, the algorithm might have poor recall.

Note that, in practice, we only use multilingual entries, which give us clusters that are sense-distinguished, as the input sense IDs (id^*) for inference. This is because an undistinguished bilingual entry may represent more than 1 sense via the same edge leading to mixing word senses after inference. We pick v_1^* to be the source word of the multilingual entry. Our task reduces to picking a second word v_2^* that is well connected and is expected to share just one sense with v_1^* from all edges (v_1^*, v) with label id^*.

To pick such a v_2^* we consider alternate senses of v_1^*, *i.e.*, id' s.t. $\langle id^*, id' \rangle \in \Psi$. We look for translations of id^* that do not appear in any id'. We prefer those languages that do appear in other id', but, with other translations. Finally, we rate the candidate words with edge degree and pick the one with the maximum connectivity as v_2^*.

3.7. Experimental results: Comparing inference algorithms

Which of the three algorithms (TRANSGRAPH, uSENSEUNIFORMPATHS and SENSEUNIFORMPATHS) is superior for translation inference? To carry out this comparison, we randomly sampled 1000 senses from English Wiktionary and ran the three algorithms over them. Our task is to compare the precision and coverage of these inference algorithms and ideally, we would like to evaluate a random sample of all the translations inferred and compare with a gold standard. Unfortunately, this kind of a comparison is virtually impossible to carry out, because of several reasons. First, gold standards for lexical translation exist for only a few language pairs. Second, they are rarely comprehensive, and may only suggest a fraction of translations instead of all. Third, they only suggest correct translations and do not specify the incorrect ones. Treating all translations absent in the gold standard to be incorrect is grossly inaccurate, since the standard does not specify all possible synonyms and the algorithms often infer synonyms in target language as valid translations. Thus we chose to employ human evaluators to determine the precision of our algorithms.

However, a high-quality evaluation of translation between two languages requires a person who is fluent in both languages. Such people are also 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 their native language as well as in English.[5]

In this experiment we evaluated the results on 7 languages – Chinese, Danish, German, Hindi, Japanese, Russian, and Turkish. We provided our informants 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. Each informant tagged 60 random translations inferred by each algorithm, which resulted in 360–400 tags per algorithm.[6] The precision over these was taken as a surrogate for the precision across all the senses.

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 (or scores for TRANSGRAPH) and compute the precision at various thresholds.

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).[7]

---

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

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

[7] Our informants tended to underestimate precision, often marking correct translations in minor senses of a word as incorrect.
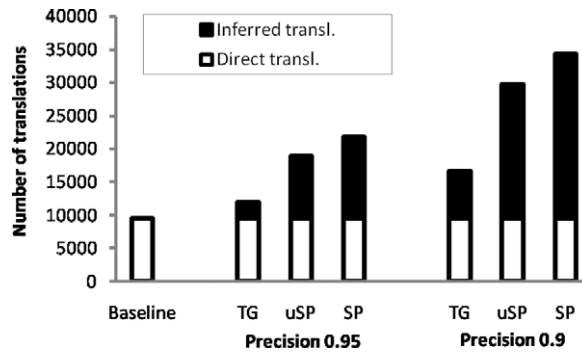
Fig. 7. The SenseUniformPaths algorithm (SP) more than doubles the number of correct translations at precision 0.95, compared to a baseline of translations that can be found without inference. The other algorithms TransGraph and uSenseUniformPaths are abbreviated as TG and uSP respectively.
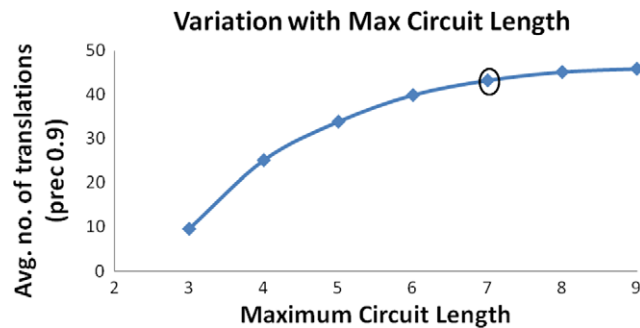

Fig. 8. The average number of translations inferred by SenseUniformPaths as a function of max circuit length for random walks. The circled point is the value used in all other experiments.

Our results are shown in Fig. 7. At this high precision, SenseUniformPaths more than doubles the number of baseline translations, finding 5 times as many inferred translations (in black) as TransGraph. The number of inferred translations (in black) for sunp is 1.2 times that of uSenseUniformPaths and 3.5 times that of TransGraph, at precision 0.9.

Indeed, both uSenseUniformPaths and SenseUniformPaths massively outperform TransGraph. SenseUniformPaths is consistently better than uSenseUniformPaths, since it performs better for polysemous words, due to its pruning based on ambiguity sets. We conclude that SenseUniformPaths is the best inference algorithm and employ it for further research.

3.8. Experimental results: Control experiments for the random walk scheme

We additionally analyze the behavior of SenseUniformPaths as a function of the parameters of the algorithm – k, the maximum length of a random walk, and N_R, the number of random walks. We randomly sampled 100 senses and ran SenseUniformPaths by keeping one variable constant and varying the other one. We report the average number of translations inferred at the probability values corresponding to precision 0.9.

Fig. 8 plots the variation of the algorithm as a function of length of the random walk. We find that around circuit length 7 the average number of translations inferred stabilize. This is also the value we picked based on the initial experiments.

Fig. 9 reports the variation as a function of the number of random walks. We observe that number of translations inferred remains on an upward trend as we increase the number of walks, though its rate slows down. Moreover, the running time of the algorithm is directly proportional to the number of walks. The current version of PanDictionary is constructed using 4000 random walks. We plan to build the next version with a larger number of walks per inference.

4. PanDictionary: A novel multilingual resource

To be most useful for our vision of panlingual translation we wish to construct a *sense-distinguished* lexical translation resource, in which each entry is a distinct word sense and associated with each word sense is a list of translations in multiple languages. This will enable lexical translation for a large number of languages at once just by looking up the desired sense. We compile PanDictionary, a first version of such a dictionary, by employing probabilistic inference over the translation graph.

The strength of SenseUniformPaths, our best inference algorithm, is that it takes a particular sense and expands it to generate a large list of translations of that sense. However, the algorithm must be supplied with the word sense, and does not have the capability to discover the different senses of a word. The strength of a manually engineered dictionary, on the
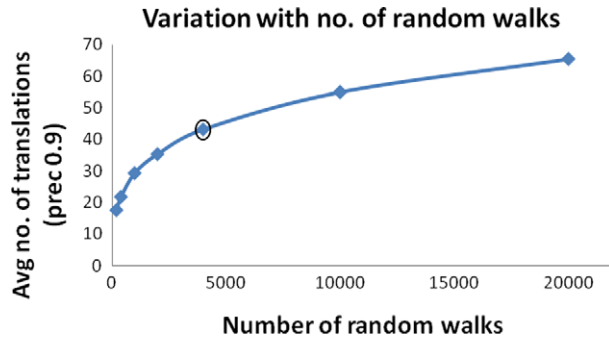
Fig. 9. The average number of translations inferred by SENSEUNIFORMPATHS as a function of number of random walks. The circled point is the value used in all other experiments.

Fig. 10. A PANDICTIONARY entry for the Croatian word 'ruža' with a portion of the 63 translations for the sense 'rose: flower'.

other hand, is a careful analysis of the senses of each word. But manual engineering limits the number of translations per word. The English Wiktionary, for example, has an average of 7.2 translations per word sense.

We exploit the synergy between Wiktionaries and the algorithm by using the senses from Wiktionaries and expanding them via SENSEUNIFORMPATHS. 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.

PANDICTIONARY can be used to look up words in any language. Fig. 10 shows a portion of an entry for the Croatian word 'ruža', which has translations for three senses: 159 translations for the color pink, 63 translations for rose the flower, and 30 for rose the shrub. This is at probability thresholds that we have found empirically to give precision of about 0.85 – higher thresholds will return fewer translations at higher precision; lower thresholds will return a much larger number of translations at lower precision.

There are other vertices, however, that may be translations of s^*, but, missed by our algorithm due to lack of evidence. In particular, there are many languages, typically the less common languages, that have only one bilingual dictionary available, usually with the closest associated common language. For example, most translations from Hawaiian are through a Hawaiian–English bilingual dictionary. These words do not have other edges to enable translation circuits.

To cater to such resource-poor languages we additionally save a set of nodes that are "singly linked" to the vertices in s^* (with a high probability). These singly linked nodes are translations of a word that is inferred to be a translation of s^*. We found empirically that words singly linked to high probability translations have the desired word sense with a precision 0.6. Unfortunately, due to limited evidence, we were unable to separate the correct translations from incorrect ones for such resource-poor languages.

In the evaluation below we investigate two key questions: (1) how does the coverage of PANDICTIONARY compare with the largest existing multilingual dictionary, the English Wiktionary (Section 4.1)? (2) what is the benefit of inference over the mere aggregation of 631 dictionaries (Section 4.2)? Additionally, we evaluate the quality of PANDICTIONARY on two other dimensions – variation with the degree of polysemy of source word, and variation with original size of the seed translation set.
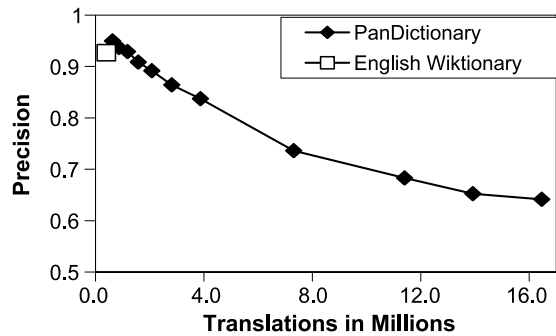
Fig. 11. Precision *vs.* coverage curve for PanDictionary. It quadruples the size of the English Wiktionary at precision 0.90, is more than 8 times larger at precision 0.85 and is almost 24 times the size at precision 0.7.

Table 1
PanDictionary covers substantially more languages than the English Wiktionary.

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

4.1. Experiments: Comparison with English Wiktionary

We first 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.[8] 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 2500 tags. Fig. 11 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 1000 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 was greater than 20.

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. Fig. 12 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 SenseUniformPaths is able to maintain adequate precision in difficult inference tasks.

4.2. Experiments: 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. The key difficulty in this evaluation arises due to the unavailability of bilingual speakers, who can speak various pairs of languages.

We evaluated the precision of word–word translations by a *collaborative tagging* scheme, with two native speakers of different languages, who are both bilingual in English. For each suggested translation they narrate in English the various

---

[8] Our translation graph uses the version of English Wiktionary extracted in January 2008.
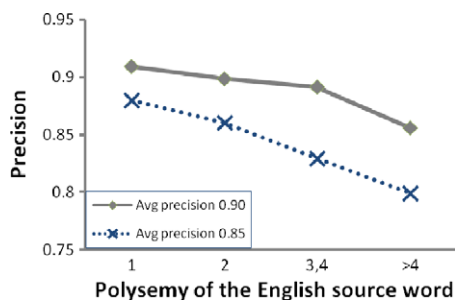
Fig. 12. Variation of precision with the degree of polysemy of the source English word. The precision decreases as polysemy increases, still maintaining reasonably high values.
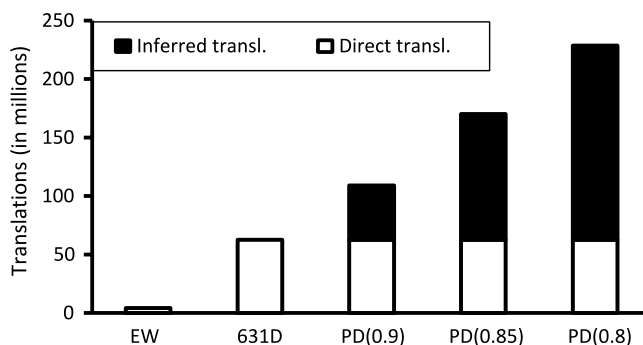

Fig. 13. The number of distinct word–word translation pairs from PanDictionary is several times higher than the number of translation pairs in the English Wiktionary (EW) or in all 631 source dictionaries combined (631 D). A majority of PanDictionary translations are inferred by combining entries from multiple dictionaries.

senses of words in their respective languages. They tag a translation correct if they found a common sense, one that is shared by both the words. For this study our informants tagged 7 language pairs: Hindi–Hebrew, Japanese–Russian, Chinese–Turkish, Japanese–German, Chinese–Russian, Bengali–German, and Hindi–Turkish. The languages were chosen based on the availability of informants and the specific pairings were randomly generated.

Fig. 13 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.

Overall, our experiments demonstrate that PanDictionary, which is our compiled dictionary, has much larger coverage than English Wiktionary, the largest multilingual dictionary known to us before this project. We also observe that our algorithms infer a large number of translations that are not in any of the input dictionaries quadrupling the number of pairwise translations asserted (at precision 0.8). This illustrates the potential impact of probabilistic inference on the construction of dictionaries and lexical resources, in general.

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.

There has been considerable research on methods to acquire translation lexicons from either MRDs [34,20,9] or from parallel text [15,14,32,13], but this has generally been limited to a small number of languages. Manually engineered dictionaries such as EuroWordNet [38] are also limited to a relatively small set of languages. There is some recent work on compiling dictionaries from monolingual corpora, which induces translations based on purely monolingual features like context counts and orthographic substrings [19]. This approach has the potential to scale to several language pairs in future.

Little work has been done in combining multiple dictionaries in a way that maintains word senses across dictionaries. Gollins and Sanderson [17] explored using triangulation between alternate pivot languages in cross-lingual information retrieval. Translating query terms from German to Dutch and then to English or translating from German to Spanish to English gave extremely low precision. The intersection of these translations did much better, although still had precision of only 0.044. They were essentially finding translation circuits from German to Dutch to English to Spanish to German.

Their triangulation essentially finds translation circuits from four bilingual dictionaries, but, unlike our SenseUniformPaths algorithm, 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 [10]. This is based on word alignments from parallel corpora that act much like bilingual dictionaries. Semantic mirrors finds all possible English translations of a Norwegian word, then all possible Norwegian translations of those words, and so forth. This forms overlapping clusters of translations that can be partitioned to discover distinct word senses. This is somewhat akin to SenseUniformPaths in that it produces a sense-distinguished dictionary from inputs that are not sense-distinguished, although its input is aligned corpora rather than dictionaries. Where SenseUniformPaths begins with a designated word sense and maintains this across multiple dictionaries, semantic mirrors fans out in all possible senses, and then clusters the results to discover senses. However, its expensive processing and reliance on parallel corpora would not scale to large numbers of languages.

Translation paths through a pivot language within a specific language family are exploited for lexicon induction by Mann and Yarowsky [29]. They use string distance models of cognate similarity, since languages within a family share many cognates. Later, this work was extended to incorporate other string distances, which are all combined for deducing the translations [35]. While promising their approach only creates bilingual lexicons, whereas our aim is to compile a sense-distinguished multilingual dictionary. In the future we wish to adapt some of their methods to benefit our translation inference within the language families.

Earlier Knight and Luk [27] 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. Other researchers worked on the sense matching problem, but with limited success [26]. Later, Schafer and Yarowsky [36] induced monolingual sense clusters and sense-hierarchy based on data from several bilingual dictionaries. In contrast, our work infers translations and also assigns them to a multilingual sense cluster.

Algorithms utilizing random walks and graph sampling techniques have become increasingly popular in the recent years (*e.g.*, [21,2]). Monte Carlo simulation is also common in estimating properties of random graphs [25]. In this paper we have adapted these techniques to work over the translation graph and its particular probabilistic semantics.

6. Applications of PanDictionary

The development of PanDictionary, a sense-distinguished global translation resource, opens exciting opportunities for applications. These applications are especially useful in reaching out to languages that are not part of common translation systems, either due to poor resources or due to lack of economic interests. Thus, applications built over PanDictionary have the potential to impact developing nations and take computing technology to far reaching areas where the technology boom hasn't had sufficient impact.

6.1. Cross lingual image search

Monolingual image search, such as Google Images, faces the challenge that most images are tagged only in resource rich languages, like English and Spanish. The number of images obtained if queried in resource poor languages is very small, making the systems limited in their global reach. Our prototype search engine, PanImages, shows lexical translations based on PanDictionary, thereby enabling them to search the same concept in different languages resulting in a much broader coverage of images. As a by product, PanImages is able to offer cross-cultural images for the same concept. For instance, searching for 'breakfast' in Dutch, Japanese and Arabic shows culturally different images of breakfast. Finally, image search based on translations also helps in situations where the original word has several meanings or has homonyms in other languages.

Currently we are developing the next generation of image search [8] by applying machine learning over PanDictionary translation sets. Our learner automatically classifies the various translations of a sense as good to query or not. The features for the learner are automatically extracted from PanDictionary and Google and reflect the expected polysemy of the translation, coverage of the language, *etc.* Given a sense for which we are interested in finding images, our system queries Google with a subset of translations recommended by our learner. Thus the onus of querying a search engine is no longer on the user (in contrast to PanImages). Our preliminary experiments show that our system finds many more relevant images compared to other systems like Google Images queried with English, or PanImages queried with a random set of translations.

6.2. Lemmatic translation and communication

With the vision of universal communication we compiled PanDictionary, which translates words between a wide array of languages. Unfortunately, the transition from translating individual words to translating sentences is non-trivial. Popular techniques rely on statistical properties of aligned corpora or a set of transfer rules constructed by language experts. Neither of these is possible at our envisioned scale. Moreover, naive ways for translating sentences using PanDictionary fall into common problems like word-sense disambiguation, and absence of morphed forms of words in PanDictionary.

While translating grammatically correct and fluent language may not be possible at this scale, we are building a novel translation system based on the hypothesis that *lemmatic communication* is enough to transmit the intended meaning of a wide variety of sentences, especially under a known context [37]. By lemmatic communication we refer to communication using only the dictionary (lemmatic) forms of a word without morphological variations, and often with inaccurate word order and missing particles. For instance, the lemmatic form for the English text "I am visiting Chicago on October 4. Do you have a room for two people?" could be "I visit Chicago October 4. Room two person?". Under the context that the recipient of the message is a hotel owner, the intended meaning of the lemmatic form is more or less clear.

We are building a translation system that takes a lemmatic message, uses manual or automatic techniques for word-sense disambiguation, and uses PanDictionary lookup to translate the message into the target language. Our preliminary results [12] show that a large fraction of messages translated in such a manner get correctly interpreted by the recipients. These results are exciting, because this method may result in a huge leap in realizing the vision of universal communication.

6.3. Plug and translate architecture

All languages lie on a continuum between existence of zero lingual resource and a huge set of monolingual and interlingual resources. At one end are languages spoken in small tribal areas, for which we probably have no or little documented resources, and on the other end are very popular languages like English and Spanish, which have a huge body of resources like thesauri, parsers, grammars, large amounts of monolingual text, dictionaries with a large number of languages and bilingual aligned corpora with several languages. However, most languages lie in the middle, where some kinds of the resources are available at varying scales and others are not.

Closer to the poor-resource end we have described a method of lemmatic communication that is able to translate simple sentences encoded in the lemmatic form. At the other end we have the full-blown statistical MT methods. We wish to explore the various middle grounds, so that as more resources become available, the quality of achievable translation for the language (language-pair) can be automatically improved. We are working on a robust architecture, which will enable language experts and other native speakers to plug in additional resources and we will be able to use those automatically to improve the quality of translation.

7. Conclusions

We have described a novel approach to the task of lexical translation, which automatically constructs a massive translation graph by parsing over 630 machine readable dictionaries and storing all asserted translations as edges in the graph. Probabilistic reasoning over the translation graph results in inferring translations that are not found in any of the source dictionaries. Using this inference procedure on different starting senses we are able to automatically construct a unique multilingual translation resource, called PanDictionary. PanDictionary is sense-distinguished and lists translations of each sense in a large number of languages (with associated confidence values).

We have developed three inference algorithms for our task, *viz.*, TransGraph, uSenseUniformPaths, and SenseUniformPaths. These exploit several insights regarding the graph topology, especially in the context of the translation graph. Our experimental comparisons show that SenseUniformPaths dominates the other two by significant margins.

We empirically evaluated PanDictionary and found that it 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. Note that our taggers evaluated the precision of English Wiktionary at 0.93, so the precision of 0.9 is close to that of the Wiktionary. Most likely, both precision numbers are underestimated due to the strictness of our evaluators. At lower precision, we are able to increase the size of the resource even more.

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.*, [22]). An exciting direction for future work is to automatically build inference rules for translation inference using labeled training data. PanDictionary is already being used for a cross-lingual image search engine. We are currently working on a machine translation system based on PanDictionary that will be capable of translating simple sentences between a large number of language-pairs.

8. Downloads

To obtain a copy of the translation graph please contact Utilika Foundation at `info@utilika.org`. For a copy of PanDictionary please email the Turing Center at `panimages@cs.washington.edu`.

Acknowledgments

experts who donated their time and language expertise to evaluate our systems. We also thank the anonymous reviewers of the drafts of the previous conference papers for their valuable suggestions in improving the evaluation and presentation.

Appendix A. Proof of Theorem 1

We prove the theorem for length $k + 2$ circuits. Let the two vertices known to share only one sense s^* be x and y. Let the intermediate vertices be $v_1, v_2, \ldots, v_K$. We wish to prove that all v_is will have sense s^* with a probability almost 1. We use induction on k to prove the theorem. For the base case $k = 0$ the theorem is vacuously true. Let us now assume that the theorem is true for $k = 0 \ldots K - 1$. We consider $k = K$ in the induction step.

Case I: There exists some node v_i in the K vertices that has sense s^*. We can now create two edges between v_i and x, as well as v_i and y. Applying induction hypothesis on the two translation circuits of smaller size we can prove that all v_is have sense s^*.

Case II: None of the intermediate nodes have sense s^*.

Case II(a): If any v_i and v_j ($j \neq i + 1$) have a sense in common then we can remove the nodes between v_i and v_j and make a shorter circuit, and use the hypothesis to prove the result. Then we can prove separately for all the discarded nodes using the hypothesis. (Similar proof holds if v_i ($i \neq 1$) has a sense common with x, or v_i ($i \neq K$) with y.)

Case II(b): The only case remains when none of x, y, v_is have any sense in common except common senses between the consecutive nodes. Let v_i has ni senses and x, y have nx and ny senses. Also let the number of senses common between v_i and v_{i+1} be $cm(i+1)$. The number of ways in which this happens is

$$\binom{|\mathcal{S}|-1}{nx-1}\binom{nx-1}{cm1}\binom{|\mathcal{S}|-nx}{n1-cm1}\binom{n1-cm1}{cm2}\binom{|\mathcal{S}|-nx-n1}{n2-cm2}\ldots\binom{n(K-1)-cm(K-1)}{cmK}\binom{|\mathcal{S}|-nx-n1-\cdots-n(K-1)}{nK-cmK}\binom{nK-cmK}{cmy}\binom{|\mathcal{S}|-nx-n1-\cdots-nK}{ny-cmy-1}.$$

The total number of ways of generating *all* sense assignments for these $K + 2$ vertices is:

$$\binom{|\mathcal{S}|-1}{nx-1}\binom{nx-1}{cm1}\binom{|\mathcal{S}|-nx}{n1-cm1}\binom{n1}{cm2}\binom{|\mathcal{S}|-n1}{n2-cm2}\ldots\binom{n(K-1)}{cmK}\binom{|\mathcal{S}|-n(K-1)}{nK-cmK}\Big[(1-\alpha)\binom{nK}{cmy}\binom{|\mathcal{S}|-nK}{ny-cmy-1}+\alpha\binom{nK-1}{cmy-1}\binom{|\mathcal{S}|-nK}{ny-cmy}\Big].$$

Here the last term (enclosed in the bracket []) refer to two cases, (i) in which v_K does not have sense s^* and (ii) in which v_K does have sense s^*. Let case (ii) happens with probability α. In that case s^* will be one of the common senses of the cmy senses and so we will only need to choose the rest $cmy - 1$ senses from nK. Moreover, since s^* will already be accounted for y can now have $ny - cmy$ more senses (as opposed to $ny - cmy - 1$ for the case (i)).

Note that, as long as α is non-zero in the limit, the product for case (ii) will dominate that for case (i), since it will have one additional product term of $O(|\mathcal{S}|)$. The probability of occurrence of Case II(b) will be dividing the two big products. Observe that all successive terms have similar orders except the last term in which the denominator has an additional $O(|\mathcal{S}|)$. Hence overall the fraction will tend to 0 as $|\mathcal{S}| \to \infty$.

Finally we need to make sure α does not tend to zero itself. Note that v_1 has s^* with probability $\frac{\binom{nx-1}{cm1-1}}{\binom{nx}{cm1}}$. If v_1 has s^* then from induction hypothesis all other nodes have s^* (Case I). Thus $\alpha > \frac{\binom{nx-1}{cm1-1}}{\binom{nx}{cm1}}$, and hence it does not tend to zero.

References

[1] E. Adar, M. Skinner, D. Weld, Information arbitrage in multi-lingual Wikipedia, in: Proc. of Web Search and Data Mining (WSDM 2009), 2009.
[2] C. Andrieu, N.D. Freitas, A. Doucet, M. Jordan, An introduction to MCMC for machine learning, Machine Learning 50 (2003) 5–43.
[3] B. Bollobas, Random Graphs, Cambridge University Press, 2001.
[4] F. Bond, S. Oepen, M. Siegel, A. Copestake, D. Flickinger, Open source machine translation with DELPH-IN, in: Open-Source Machine Translation Workshop at MT Summit X, 2005.
[5] P. Brown, S.D. Pietra, V.D. Pietra, R. Mercer, The mathematics of machine translation: parameter estimation, Computational Linguistics 19 (2) (1993) 263–311.
[6] J. Carbonell, S. Klein, D. Miller, M. Steinbaum, T. Grassiany, J. Frey, Context-based machine translation, in: AMTA, 2006.
[7] D. Chiang, A hierarchical phrase-based model for statistical machine translation, in: ACL, 2005.
[8] J. Christensen, Mausam, O. Etzioni, A rose is a roos is a ruusu: Querying translations for web image search, in: ACL'09, 2009.
[9] A. Copestake, T. Briscoe, P. Vossen, A. Ageno, I. Castellon, F. Ribas, G. Rigau, H. Rodriquez, A. Samiotou, Acquisition of lexical translation relations from MRDs, Machine Translation 3 (3–4) (1994) 183–219.
[10] H. Dyvik, Translation as semantic mirrors: from parallel corpus to WordNet, Language and Computers 49 (1) (2004) 311–326.
[11] O. Etzioni, K. Reiter, S. Soderland, M. Sammer, Lexical translation with application to image search on the Web, in: Machine Translation Summit XI, 2007.
[12] K. Everitt, C. Lim, O. Etzioni, J. Pool, S. Colowick, S. Soderland, Evaluating lemmatic communication, in: trans-kom 3, 2010.
[13] M. Franz, S. McCarly, W. Zhu, English–Chinese information retrieval at IBM, in: Proceedings of TREC 2001, 2001.
[14] P. Fung, A pattern matching method for finding noun and proper noun translations from noisy parallel corpora, in: Proceedings of ACL-1995, 1995.
[15] W. Gale, K. Church, A program for aligning sentences in bilingual corpora, in: Proceedings of ACL-1991, 1991.
[16] A.V. Goldberg, S. Rao, Beyond the flow decomposition barrier, Journal of the ACM 45 (5) (1998) 783–797.
[17] T. Gollins, M. Sanderson, Improving cross language retrieval with triangulated translation, in: SIGIR, 2001.
[18] R.G. Gordon Jr. (Ed.), Ethnologue: Languages of the World, fifteenth edition, SIL International, 2005.
[19] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, D. Klein, Learning bilingual lexicons from monolingual corpora, in: ACL, 2008.
[20] S. Helmreich, L. Guthrie, Y. Wilks, The use of machine readable dictionaries in the Pangloss project, in: AAAI Spring Symposium on Building Lexicons for Machine Translation, 1993.
[21] M.R. Henzinger, A. Heydon, M. Mitzenmacher, M. Najork, Measuring index quality using random walks on the web, in: WWW, 1999.

[22] R. Hoffmann, S. Amershi, K. Patel, F. Wu, J. Fogarty, D.S. Weld, Amplifying community content creation with mixed-initiative information extraction, in: ACM SIGCHI (CHI 2009), 2009.
[23] D. Hull, G. Grefenstette, Querying across languages: a dictionary-based approach to multilingual information retrieval, in: Proceedings of ACM SIGIR, 1996, pp. 49–57.
[24] W.J. Hutchins (Ed.), Machine Translation: Past, Present, Future, Halted Press, New York, 1986.
[25] D.R. Karger, A randomized fully polynomial approximation scheme for the all-terminal network reliability problem, SIAM Journal of Computation 29 (2) (1999) 492–514.
[26] J. Klavans, E. Tzoukermann, The BICORD system: Combining lexical information from bilingual corpora and machine readable dictionaries, in: COLING, 1990.
[27] K. Knight, S. Luk, Building a large-scale knowledge base for machine translation, in: AAAI, 1994.
[28] P. Koehn, F.J. Och, D. Marcu, Statistical phrase-based translation, in: HLT/NAACL, 2003.
[29] G. Mann, D. Yarowsky, Multipath translation lexicon induction via bridgle languages, in: NAACL, 2001.
[30] J. Martin, R. Mihalcea, T. Pedersen, Word alignment for languages with scarce resources, in: ACL Workshop on Building and Using Parallel Text, 2005.
[31] Mausam, S. Soderland, O. Etzioni, D. Weld, M. Skinner, J. Bilmes, Compiling a massive, multilingual dictionary via probabilistic inference, in: ACL'09, 2009.
[32] I. Melamed, A word-to-word model of translational equivalence, in: Proceedings of ACL-1997 and EACL-1997, 1997, pp. 490–497.
[33] R. Moore, A discriminative framework for bilingual word alignment, in: HLT/EMNLP, 2005, pp. 81–88.
[34] M. Neff, M. McCord, Acquiring lexical data from machine-readable dictionary resources for machine translation, in: 3rd Int. Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, 1990.
[35] C. Schafer, D. Yarowsky, Inducing translation lexicons via diverse similarity measures and bridge languages, in: CONLL, 2002.
[36] C. Schafer, D. Yarowsky, Exploiting aggregate properties of bilingual dictionaries for distinguishing senses of English words and inducing English sense clusters, in: ACL, 2004.
[37] S. Soderland, C. Lim, Mausam, B. Qin, O. Etzioni, J. Pool, Lemmatic machine translation, in: MT Summit XII, 2009.
[38] P. Vossen (Ed.), EuroWordNet: A Multilingual Database with Lexical Semantic Networks, Kluwer Academic Publishers, 1998.