Statistical Machine Translation without Parallel Corpora

Person
University of Awesome
Location
person@email

Person
University of Awesome
Location
person@email

Person
University of Awesome
Location
person@email

Abstract

1 Introduction

2 Related work

SMT state of the art: phrase-based (Koehn et al., 2003), hierarchical (Chiang, 2005), etc. Will review the phrase-based pipeline in greater detail in Section 3, and describe our extensions in Section 4. For a recent thorough review of statistical machine translation we refer the reader to (Lopez, 2008).

A lot of related work in lexicon induction (e.g. (Rapp, 1995; Fung and Yee, 1998; Koehn and Knight, 2000; Haghighi et al., 2008; Mimno et al., 2009)). Although often motivated by the resource constrained machine translation, it was never used in MT. In this work, we make direct use of these methods in the machine translation pipeline.

3 Background

Review the MT pipeline, emphasizing: phrase scoring and ordering.

4 Reducing the Parallel Data requirement / Estimating Parameters from Monolingual Data

4.1 Phrase extraction

Building on the many successful efforts in bilingual lexicon induction from monolingual corpora, we compile a table of *phrasal* translations from monolingual corpora. Current methods for inducing a bilingual lexicon, or, equivalently, a unigram phrase translation table, are computationally expensive as each source and target word pair must be scored. That is, the models must compute $|V_s| * |V_t|$ similarity scores, where $|V_s|$ and $|V_t|$ are the sizes of the source and target vocabularies, respectively. As we search for longer phrase pairs, this search space increases exponentially with the size of the n-grams. Exhaustively scoring all phrase pairs up to length three requires $|V_s|^3 * |V_t|^3$ score computations. We prune the phrase pair search space using methods from Information Retrieval, like that used in Uszkoreit et al. (2010).

First, we compose a set of all n-grams up to length three in the source side monolingual corpus and in the target side monolingual corpus. We store each source side n-gram's frequency along with the phrase. We also store each target side n-gram in a frequency inverted index. We index target side phrases by frequencies within a band of their actual observed frequency in the monolingual corpus. Additionally, for the target side n-grams, we look up each word in the phrase in our bilingual dictionary, and we store a set of source side words that translate into any word in the phrase.

discuss dictionary(ies)

These sets of observed phrases, frequencies, and the bilingual dictionary together allow us to effectively filter our search space using combinations of inverted indices. Given a source side test set, we collect n-grams up to length three (in our Urdu test set, there are 36,423 phrases), as we did for the monolingual corpora. We then look up each source side phrase's monolingual corpus frequency and consult the target side inverted index to find a list of target side phrases that occur in the same frequency band. Of those target side phrases, we keep only the ones with at least one word translating into a word in the source side phrase.

discuss dictionary parameters

This method of pruning the phrase pair search space involves two manually tuned parameters. In order to evaluate the filtered phrase tables and tune the parameters, we used the Moses decoder's trace function to find the set of phrases used in decoding an Urdu test set. We compare our filtered phrase tables to this set of phrase translation rules and at-

data details?

tempt to maintain as many of them as possible, while pruning the set of phrase pairs down to manageable size.

Our baseline phrase table is generated using a bilingual dictionary. For each Urdu test set phrase up to length three, we generated English phrases from all combinations of dictionary translations and all possible reorderings. For the baseline and our pruning methods, the number of filtered phrase pairs and the percent of phrases used by the Moses decoder not pruned away are given in Table 4.1.

Second round of pruning: after monolingual feature extraction, before re-ordering estimation. Needs to be discussed after explanation of those methods?

4.2 Phrase scoring

In place of phrase translation probabilities estimated from bilingual alignments, we propose to compute similarity scores computed (almost) solely from monolingual resources.

Contextual similarity. We extend the vector space approach of (Rapp, 1999) to compute similarity between *phrases* in source and target language. More formally, assume that $(f_1, f_2, \ldots f_N)$ and $(e_1, e_2, \ldots e_M)$ are (arbitrarily indexed) source and target vocabularies, respectively. A source phrase f (target phrase e) is represented with an N (M) dimensional vector. Only the components corresponding to words that appear in the context of f (e) in data take on non-zero values, which typically measure how "unique" a word is to the context in the dataset. Next, f's contextual vector is projected by mapping each component to a component in the target space corresponding to its translation (taken from a small seed dictionary), but retaining the source component value. Finally, the pair (f, e) is scored by computing similarity between the (projected) source and target vectors. Various means of computing the component values and vector similarity measures have been proposed in literature (e.g. (Rapp, 1999; Fung and Yee, 1998)). While the quality of the resulting induced lexicon depends on the data, we found the following to work best in our experiments. We compute the value of the k-th component of f's contextual vector as follows:

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

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

Temporal similarity. Online content is often published along with temporal information: news feeds, for example, are comprised of news stories annotated with date and time of publication. The feeds are specialized for the target geographical locations and vary in content across languages. Still, many events are deemed relevant to multiple audiences and the news stories related to them appear in several languages, although rarely as direct translations of one another. Phrases associated with these events will appear with increased frequency in multiple languages around the dates when these events are reported. Such weak synchronicity provides a cue about the relatedness of phrases across the two languages. In order to score a pair of phrases across languages, we can compute the similarity of their temporal signatures. To generate a time sequence for a given word, we first sort the set of (time-stamped) documents of our corpus into a sequence of equally sized temporal bins. We then count the number of occurrences of a phrase in each bin. Changing the size of the bin or computing counts in a sliding window instead can recover some accuracy if the temporal alignment between two languages in our dataset is poor (Klementiev and Roth, 2006). Finally, we normalize the sequence and use either the cosine measure to score similarity.

Orthographic / phonetic similarity. Etymologically related words often retain similar spelling across languages with the same writing system, and the edit distance can be used to measure their orthographic similarity. We extend this idea to phrases by using word alignments within a phrase pair (see Section 4.1): we score pairs of aligned words and normalize by their average length.

Make sure it is correct.

We can further extend this idea to language pairs not sharing the same writing system, since many cognates and transliterated words are phonetically similar. Following (Virga and

Pruning filters	Phrase Pairs	Percent of total search space	Findable types	Findable tokens
Unpruned phrase table	37,322,465,985	100%	100%	100%
Baseline phrase pairs	29245036	0.08%	15.37	25.07
Frequency-based pruning	4,450,429,494	11.92%	85.21%	87.48%
Frequency and Dictionary pruning	1,436,823,109	3.85%	57.79%	58.76%

Table 1: This shows the tradeoff between pruning the phrase pair search space and the accuracy of the final set of phrase pairs. The findable types and tokens measures refer to the percent of phrase types and tokens used by Moses to decode a test set that are not pruned away.

Khudanpur, 2003; Irvine et al., 2010), we treat transliteration as a monotone character translation task and use a generative model to propose a transliteration of tokens in a source phrase. Once the source words are mapped to the target writing system, the phrase similarity is computed as before.

Argue that enough training data is easy to get

Depending on the available monolingual data (and its associated metadata), various other similarity scores can be computed and added to the list (see, e.g. (Schafer and Yarowsky, 2002)).

4.3 Reordering

TODO: Reordering figure

Describe the algorithm for estimating orientation probabilities. Talk about the issue of too much weight on out-of-order orientation. Talk about where w comes from.

5 Experiments

5.1 Data

Describe data we use in the experiments: Europarl (Koehn, 2005), Gigaword, our own crawls[1].

5.2 Single language

1. *Phrase features.* (a) Augment phrase scores with mono features. If we see better performance, reduce the amount of parallel data until it matches the performance of the original system. Make the tradeoff argument. (b) (**lesion experiments**) See how well we do with mono features alone.

2. *Orientation features.* Use mono orientation features.

3. *Induce phrase table.*

4. *Put everything together.* Run the entire pipeline.

```
Input: Source and target phrases $f$ and $e$,
       source and target corpora $C_f$ and $C_e$.
Output: Reordering features $(p_m, p_s, p_o)$.
Find all sentences $S_f$ containing $f$ in $C_f$;
Find all sentences $S_e$ containing $e$ in $C_e$;
$c_m = c_s = s_o = 0$;
foreach $S_f$ do
    foreach phrase $f_p$ preceding $f$ in $S_f$ do
        foreach translation $e^*$ of $f_p$ do
            foreach $S_e$ containing $e^*$ do
                $w = p(e^*|f)$;
                if $e^*$ precedes $e$ in $S_e$ then
                    $c_m += w$;
                else if $e^*$ follows $e$ in $S_e$ then
                    $c_s += w$;
                else
                    $c_o += w$;
                c += w;
return $(\frac{c_m}{c}, \frac{c_s}{c}, \frac{c_o}{c})$
```

Figure 1: Estimating reordering probabilities from monolingual data.

5.3 Big experiment

Now, run the entire pipeline on a handful of languages extracting monolingual features from the Gigaword and our crawls.

6 Discussion

7 Conclusions and Future Work

First to make use of plentiful monolingual data to reduce the dependence on expensive parallel data. In particular:

- Showed that augmenting standard pipeline with monolingual features helps.

- Demonstrated that monolingual features are informative enough on their own for a competitive system.

[1] Promise to distribute after publication.

- Proposed an algorithm for estimating orientation probabilities from monolingual data alone.
- Build complete systems for X low-resource languages.

References

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

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

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ann Irvine, Chris Callison-Burch, and Alexandre Klementiev. 2010. Transliterating from all languages. In *Proc. of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

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

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

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 48–54.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. of the Machine Translation Summit*.

Adam Lopez. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3):1–49, August.

David Mimno, Hanna Wallach, Jason Naradowsky, David Smith, and Andrew McCallum. 2009. Polylingual topic models. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

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

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

Jakob Uszkoreit, Jay M. Ponte, Ashok C. Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proc. of the International Conference on Computational Linguistics (COLING)*.

Paola Virga and Sanjeev Khudanpur. 2003. Transliteration of proper names in cross-lingual information retrieval. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.