Statistical Structure in Language Processing
Phrase based models

Cristina Gârbacea
10407936
cr1st1na.garbacea@gmail.com

Sara Veldhoen
10545298
sara.veldhoen@student.uva.nl

Abstract

Currently in statistical machine translation the best performing systems rely on phrase pairs extracted from word aligned parallel corpora. The number of possible phrase pairs in parallel data is huge, so this space needs to be reduced for computational reasons. We take the common heuristic approach to select only those pairs that are consistent with word-aligned data. We implemented a phrase pair extraction algorithm and run it with alignments provided by *Giza++*. We also present how to get the conditional and joint probability estimates of the phrases. We trained and evaluated Moses, an existing phrase-based translation toolkit. We compare our phrase table to the phrases extracted by Moses, computing precision and recall . Finally, we explore the coverage of our phrase table with respect to held-out data, and report on the results we obtained, along with possible improvements to our approach.

1 Introduction

In this paper we explore the utility of phrase based models inside a statistical machine translation system. As compared to our previous assignment where we only used word aligned models, in this assignment our goal is to build an efficient phrase pair extraction tool that would extract phrase pairs of up to length 4 from a given word-aligned parallel training corpus.

Since word based models translate words as atomic units, they fall short of capturing dependencies between groups of words. Even more so in cases where each source word is aligned to exactly one target word. Phrase based models can overcome this limitation by treating phrases, sequences of words, as atomic translation units, in order to make use of local context in the translation process. Phrase based translation models give improved translations over the IBM models and are capable of giving state-of-the-art translations for many pairs of languages.

We implemented a phrase extraction algorithm that we compare to Moses, an existing statistical machine translation system that allows for automatically training translation models for any language pair. We introduce Moses and the evaluation metric we use, Bleu, in section 2. In section 3 follows a presentation of the phrase extraction algorithm we implemented and we offer an insight into how to get joint and conditional probability estimates. In section 4 we present the results we obtained and finally we conclude in section 5.

2 Related Work

As outlined in (5), phrase based statistical machine translation has emerged nowadays as the dominant paradigm in machine translation research. *Moses* (4) is an open source toolkit which contains tools for data processing and training language and translation models, as well as components for tuning these models and evaluation of the translated output. In order to avoid duplicates, *Moses* makes use of *Giza++* (6) for word alignments and *SRILM* (7) for language modeling.

The training pipeline and the decoder are the main components of the system. Parallel data is fed as input to the training process (typically preprocessed beforehand by ignoring long sentences and removing misaligned sentence pairs) and the translation correspondences between the two languages are inferred by determining word co-occurences. The training procedure is split into 9 distinct steps: after preparing the data, word alignments are taken from the intersection of bidirectional rules of *Giza++* and some alignment points from the union of the two runs, based on a number of heuristics. In this manner it is pos-

sible to estimate the maximum likelihood translation table useful for phrase extraction. The extracted phrases are then scored by collecting counts and computing the phrase translation probabilities $\phi(e|f)$ for each foreign phrase f. A reordering model based on distance and a generation model built from the target side of the parallel corpus are both included in the final configuration file which is generated last and needed for decoding.

Given a trained translation model and a source sentence, the decoder will translate the source sentence into the target language by finding the highest scoring sentence in the target language which matches the source sentence. The Moses decoder uses the beam search algorithm that quickly finds the highest probability translation among an exponential number of choices. The score for a translation is computed using the weights of the individual phrases it contains (by looking them up in the Moses phrase table) and the overall language model probability of the combination (by querying the language model for the target language). Hypothesis that cover all foreign words are the ones kept in the final state.

BLEU The resulting translations are evaluated using the *BLEU* metric to assess how close the machine translation is to one or more reference translations. In our case, we use one human translation as reference. Scores are calculated for each sentence and then averaged over the whole corpus, which basically means that the more reference translations per sentence, the higher the score will be. Averaging individual sentence judgment errors over a test corpus, instead of attempting to find the exact human judgment for every sentence, confers reliability to this metric and makes it correlate highly with human judgments.

3 Phrase Extraction and weight estimation

In this section, we present our approach to the extraction of phrase pairs from the corpus. We also compute coverage of the phrase table. The estimation of translation probabilities is done in both a conditional and joint fashion.

Phrase Extraction The number of possible phrase pairs per sentence pair is huge: each sentence can be partitioned in a vast amount of ways, and each partition could form a phrase pair with any partition in the paired sentence.

In order to reduce the space, we consider only phrase pairs that are consistent with word alignments produced by IBM models. As in (1), consistency is defined as follows:

$$\begin{aligned}\langle \bar{e}, \bar{f} \rangle \text{ is consistent with } A \Leftrightarrow \\ \forall e_i \in \bar{e} : \langle e_i, f_j \rangle \in A \Rightarrow f_j \in \bar{f}, \\ \text{and } \forall f_j \in \bar{f} : \langle e_i, f_j \rangle \in A \Rightarrow e_i \in \bar{e}, \\ \text{and } \exists e_i \in \bar{e}, f_j \in \bar{f} : \langle e_i, f_j \rangle \in A.\end{aligned}$$

For this assignment, the symmetrized alignments of the corpus sentences were given. We base our extraction algorithm on the one presented in (1, page 133). We iterate over all windows up to a certain length in the English sentence, and find the foreign windows that are consistent given the alignment. For all valid pairs of windows, we extract the corresponding phrase pair. Note that we keep counts of the thus extracted phrase pairs, for efficient weight estimation in a later stage.

Coverage of the phrase table To investigate the usefulness of the extracted phrases for unknown text, we compute the coverage as compared to held-out data: a similar but independent parallel corpus. For this purpose, we run the same extraction algorithm on the held-out data. Then, we check for each extracted phrase pair whether it is in the phrase table. If it is not, however, we might be able to construct the phrase pair from phrases in the phrase table. This is a hard problem: each phrase in the held-out phrase pair of length n can be split in 2^{n-1} ways. Moreover, any phrase part in the source can be aligned to any phrase part in the target, thus introducing a combinatoric problem. Fortunately, we restricted the extracted phrases to a length of 4, so that the number of parts that we need to combine is at most $4 * 4$, with $4!$ possible combinations.

First, we create the possible splits for both the input and output phrase. Those are the input to the constructing algorithm. Since the phrase table is stored as a mapping from target to source phrases, we start from the target side. We begin with one possible target split, and look up the leftmost phrase part in the table. If it translates to any part of the foreign phrase, we recursively call the building algorithm with the remainder of the target and source phrase. Whenever a target part cannot be translated, the search for this split option is abandoned.

The coverage percentage is the relative number of phrase pairs in the heldout set that are either in the phrase table or can be constructed as described above.

Conditional Probability Estimates After having extracted the phrase pairs, we compute the conditional translation probability estimates for a foreign phrase $\overline{f}$ given an English phrase $\overline{e}$, using the following formula:

$$\phi(\overline{f}|\overline{e}) = \frac{count(\overline{e}, \overline{f})}{\sum_{\overline{f}_i} count(\overline{e}, \overline{f}_i)}$$

Here $count(\overline{e}, \overline{f})$ denotes in how many sentence pairs a specific phrase occurs and is extracted. To get relative frequency estimates, we normalize this value by the count of occurences of all phrase pairs containg the English phrase $\overline{e}$ inside the whole corpus.

Joint Probability Estimates In (2) quite a different approach is taken to phrase based translation. The idea of a noisy channel, that a foreign sentence is a corrupted version of an original English sentence, is abandoned. Rather, the two sentences are considered different substantiation of a bag of concepts. In this framework, the probability of a phrase pair is a joint probability conditioned on a concept. In practice, we do not explicitly model the concept but view the phrase pair itself as a concept, so its weight is just the joint translation probability of the two phrases: $t(\bar{e}, \bar{f})$.

The estimation of the translation probabilities in (2) is done in an adapted version of expectation maximization. In our implementation, we do not consider all alignments of all possible phrases, but instead base the extraction of phrases on the symmetrized word alignments from IBM models. Therefore, we can gather the counts of these phrase pairs directly:

$$\begin{aligned}\phi(\overline{f}, \overline{e}) &= \phi(\overline{f}|\overline{e}) \times \phi(\overline{e}) \\ &= \frac{count(\overline{e}, \overline{f})}{\sum_{\overline{f}_i, \overline{e}_i} count(\overline{e}, \overline{f}_i)}\end{aligned}$$

4 Experiments and Results

Test set results Using the data provided to us for training consisting of 100K Dutch-English bilingual sentences, we used the *Moses* implementation to train a translation system. Word-alignments were obtained by running *MGiza++*, since it supports multi-threading which speeds up the alignment process. Phrase extraction and scoring, lexicalized reordering tables and the *Moses* configuration file were created automatically by running the `train-model.perl` script. After the training process completed, we tested the decoder on the given test data consisting of 2.000 Dutch sentences. Finally we evaluated our results by running the *BLEU* script `multi-bleu.perl` script. The results are presented in Table 1:

Bleu Score	24.68
1-gram precision	59.5
2-gram precision	31.4
3-gram precision	19.0
4-gram precision	11.8
BP	0,970
ratio	0,971
hyp_len	49.018
ref_len	50.488

Table 1: *BLEU* Results on the test set, where the following abbreviations stand for: BP (brevity penalty) = min(1, number of output words / number of reference words, ratio = number of output words / number of reference words, hyp_len = number of output words, ref_len = number of reference words

Phrase table coverage The training data consisted of 100,000 parallel sentences, the held-out corpus had 2000 sentences. Out of 95,073 consistent phrase pairs in the held-out corpus, 31.5% could be constructed using the phrase table. Intuitively, this is quite a bad performance. As many as 95% of these were not directly extracted from the phrase table, but built from smaller parts.

We analyzed which phrase pairs would be built. We observed that the vast majority consisted of phrases with punctuation. Naturally, punctuation is often aligned in the symmetrized word alignments. The phrase extraction algorithm behaves greedily, in that it adds as much as possible to a consistent phrase pair, including punctuation. Although sequences of word can often contribute to meaningful phrase pairs, adding punctuation mainly appears to introduce a lot of data sparsity.

Recall and precision We further calculated Precision and Recall figures of the phrases we extracted with respect to the phrases extracted by Moses. The size of the phrase table outputted by Moses is 7,277,021, our algorithm extracted 3,901,266 phrase pairs.

The alignments given to us for this assignment were based on a different data set. Therefore, high precision and recall numbers are not to be expected. For this reason, we also run our extraction algorithm with the alignments from Moses. The size of our phrase table is roughly the same: 3,904,713 Both results are shown in Table 2. The maximum default length of the phrases extracted by Moses is set to 7, whereas our default is 4. We also ran our extraction algorithm with the maximum length equal to that of Moses, thus obtaining 1.0 precision and recall, which confirms the reliability of our approach.

Case	Precision	Recall
Given alignments	0.754	0.404
Moses alignments	1.0	0.537

Table 2: Precision and Recall of the extracted phrases with respect to the phrases extracted by Moses

5 Conclusion

Although phrase based translation has proved its worth, the attention of the field of statistical machine translation has shifted from the simple approach presented here towards new and even better models using for instance hierarchical or syntactical phrases. The phrase tables are built in a more sophisticated way.

An approach that we are curious to investigate, is to improve the phrase table quality by using evidence from more than two languages, for instance by triangulation. This would be an extension to the phrase based translation to take it to a new level.

Furthremore, it might be interesting to explore the role of punctuation in translation. Of course, punctuation occurs a lot in textual data, but it is hardly ever sensible to build phrases with punctuation. As mentioned in paragraph 4, we noted that many of the phrases in the held-out set that needed to be constructed from smaller phrases, dealt with punctuation. The size of phrase tables could possibly be reduced by treating punctuation in a special way in the extraction stage.

References

[1] Philipp Koehn, 2010. *Statistical Machine Translation*. Cambridge University Press.

[2] Daniel Marcu and William Wong, 2002. *A phrase-based, joint probability model for statistical machine translation*. Association for Computational Linguistics.

[3] Franz Josef Och, Christoph Tillmann, and Hermann Ney, 1999. *Improved alignment models for statistical machine translation* Proceedings of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora.

[4] Koehn, Philipp, 2014. *Statistical Machine Translation System. User Manual and Code Guide* `http://statmt.org/moses/manual/manual.pdf`.

[5] Koehn, Philipp, et al 2007 *Moses: Open source toolkit for statistical machine translation*. Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions.

[6] Franz Josef Och and Hermann Ney 2003 *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, 1:29, 19-51.

[7] Andreas Stolcke 2002 *SRILM - An Extensible Language Modeling Toolkit*. 901–904.

[8] Kishore Papineni and Salim Roukos and Todd Ward and Wei-Jing Zhu 2002 *BLEU - A Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the Association for ACL. 311–318.