Extending MARIE: an N-gram-based SMT decoder

Josep M. Crego
TALP Research Center
Universitat Politècnica de Catalunya
Barcelona, 08034
jmcrego@gps.tsc.upc.edu

José B. Mariño
TALP Research Center
Universitat Politècnica de Catalunya
Barcelona,08034
canton@gps.tsc.upc.edu

Abstract

In this paper we present several extensions of MARIE[1], a freely available N-gram-based statistical machine translation (SMT) decoder. The extensions mainly consist of the ability to accept and generate word graphs and the introduction of two new N-gram models in the log-linear combination of feature functions the decoder implements. Additionally, the decoder is enhanced with a caching strategy that reduces the number of N-gram calls improving the overall search efficiency. Experiments are carried out over the Eurpoean Parliament Spanish-English translation task.

1 Introduction

Research on SMT has been strongly boosted in the last few years, partially thanks to the relatively easy development of systems with enough competence as to achieve rather competitive results. In parallel, tools and techniques have grown in complexity, which makes it difficult to carry out state-of-the-art research without sharing some of this toolkits. Without aiming at being exhaustive, GIZA++[2], SRILM[3] and PHARAOH[4] are probably the best known examples.

We introduce the recent extensions made to an N-gram-based SMT decoder (Crego et al., 2005), which allowed us to tackle several translation issues (such as reordering, rescoring, modeling, etc.) successfully improving accuracy, as well as efficiency results.

As far as SMT can be seen as a double-sided problem (modeling and search), the decoder emerges as a key component, core module of any SMT system. Mainly, any technique aiming at dealing with a translation problem needs for a decoder extension to be implemented. Particularly, the reordering problem can be more efficiently (and accurate) addressed when tightly coupled with decoding. In general, the competence of a decoder to make use of the maximum of information in the global search is directly connected with the likeliness of successfully improving translations.

The paper is organized as follows. In Section 2 we and briefly review the previous work on decoding with special attention to N-gram-based decoding. Section 3 describes the extended log-linear combination of feature functions after introduced the two new models. Section 4 details the particularities of the input and output word graph extensions. Experiments are reported on section 5. Finally, conclusions are drawn in section 6.

2 Related Work

The decoding problem in SMT is expressed by the next maximization: $\arg\max_{t_1^I \in \tau} P(t_1^I|s_1^J)$, where s_1^J is the source sentence to translate and t_1^I is a possible translation of the set τ, which contains all the sentences of the language of t_1^I.

Given that the full search over the whole set of target language sentences is impracticable (τ is an infinite set), the translation sentence is usually built incrementally, composing partial translations of the source sentence, which are selected out of a limited number of translation candidates (translation units).

The first SMT decoders were **word-based**. Hence, working with translation candidates of single source words. Later appeared the **phrase-based** decoders, which use translation candidates composed of sequences of source and target words (outperforming the word-based decoders by introducing the word context). In the last few years **syntax-based** decoders have emerged aiming at dealing with pair of languages with different syntactical structures for which the word context introduced

[1]http://gps-tsc.upc.es/soft/soft/marie
[2]http://www.fjoch.com/GIZA++.html
[3]http://www.speech.sri.com/projects/srilm/
[4]http://www.isi.edu/publications/licensed-sw/pharaoh/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 213–216,
Prague, June 2007.

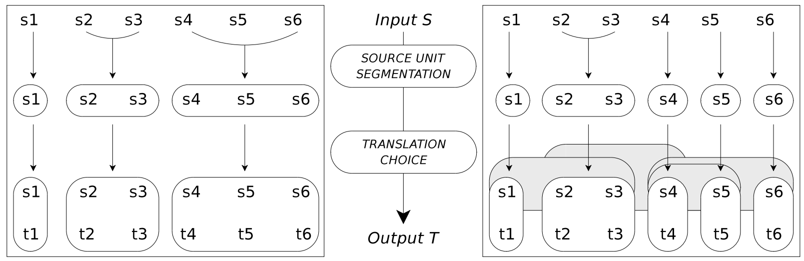

Figure 1: *Generative process. Phrase-based (left) and N-gram-based (right) approaches.*

in phrase-based decoders is not sufficient to cope with long reorderings.

Like standard phrase-based decoders, MARIE employs translation units composed of sequences of source and target words. In contrast, the translation context is differently taken into account. Whereas phrase-based decoders employ translation units uncontextualized, MARIE takes the translation unit context into account by estimating the translation model as a standard N-gram language model (N**-gram-based** decoder).

Figure 1 shows that both approaches follow the same generative process, but they differ on the structure of translation units. In the example, the units *'s1#t1'* and *'s2_s3#t2_t3'* of the N-gram-based approach are used considering that both appear sequentially. This fact can be understood as using a longer unit that includes both (longer units are drawn in grey).

MARIE follows the maximum entropy framework, where we can define a translation hypothesis t given a source sentence s, as the target sentence maximizing a log-linear combination of feature functions:

$$\hat{t}_1^I = \arg\max_{t_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^J, t_1^I) \right\} \quad (1)$$

where λ_m corresponds to the weighting coefficients of the log-linear combination, and the feature functions $h_m(s,t)$ to a logarithmic scaling of the probabilities of each model. See (Mariño et al., 2006) for further details on the N-gram-based approach to SMT.

3 N-gram Feature Functions

Two language models (LM) are introduced in equation 1, aiming at helping the decoder to find the right translations. Both are estimated as standard N-gram LM.

3.1 Target-side N-gram LM

The first additional N-gram LM is destinated to be applied over the target sentence (tagged) words. Hence, as the original target LM (computed over raw words), it is also used to score the fluency of target sentences, but aiming at achieving generalization power through using a more generalized language (such as a language of Part-of-Speech tags) instead of the one composed of raw words. Part-Of-Speech tags have successfully been used in several previous experiments. however, any other tag can be applied.

Several sequences of target tags may apply to any given translation unit (which are passed to the decoder before it starts the search). For instance, regarding a translation unit with the english word '*general*' in its target side, if POS tags were used as target tagged tags, there would exist at least two different tag options: *noun* and *adjective*.

In the search, multiple hypotheses are generated concerning different target tagged sides (sequences of tags) of a single translation unit. Therefore, on the one side, the overall search is extended towards seeking the sequence of target tags that better fits the sequence of target raw words. On the other side, this extension is hurting the overall efficiency of the decoder as additional hypotheses appear in the search stacks while not additional translation hypotheses are being tested (only differently tagged).

This extended feature may be used toghether with a limitation of the number of target tagged hypotheses per translation unit. The use of a limited number of these hypotheses implies a balance between accuracy and efficiency.

3.2 Source-side N-gram LM

The second N-gram LM is applied over the input sentence tagged words. Obviously, this model only makes sense when reordering is applied over the source words in order to monotonize the source and target word order. In such a case, the tagged LM is learnt over the training set with reordered source words.

Hence, the new model is employed as a reordering model. It scores a given source-side reordering hypothesis according to the reorderings made in the training sentences (from which the tagged LM is estimated). As for the previous extension, source tagged words are used instead of raw words in order to achieve generalization power.

Additional hypotheses regarding the same translation unit are not generated in the search as all input sentences are uniquely tagged.

Figure 2 illustrates the use of a source POS-tagged N-

gram LM. The probability of the sequence '*PRN VRB NAME ADJ*' is greater than the probability of the sequence '*PRN VRB ADJ NAME*' for a model estimated over the training set with reordered source words (with english words following the spanish word order).

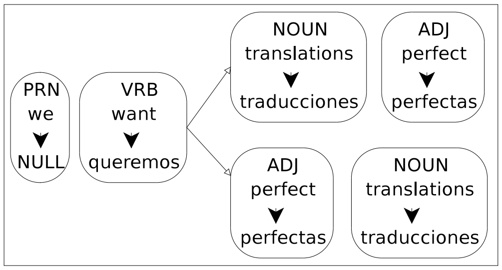

Figure 2: *Source POS-tagged N-gram LM.*

3.3 Caching N-grams

The use of several N-gram LM's implies a reduction in efficiency in contrast to other models that can be implemented by means of a single lookup table (one access per probability call). The special characteristics of Ngram LM's introduce additional memory access to account for backoff probabilities and lower Ngrams fallings.

Many N-gram calls are requested repeatedly, producing multiple calls of an entry. A simple strategy to reduce additional access consists of keeping a record (**cache**) for those Ngram entries already requested. A drawback for the use of a cache consists of the additional memory access derived of the cache maintenance (adding new and checking for existing entries).

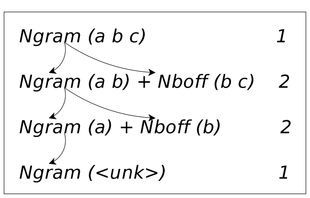

Figure 3: *Memory access derived of an N-gram call.*

Figure 3 illustrates this situation. The call for a 3-gram probability (requesting for the probability of the sequence of tokens '*a b c*') may need for up to 6 memory access, while under a phrase-based translation model the final probability would always be reached after the first memory access. The additional access in the N-gram-based approach are used to provide lower N-gram and backoff probabilities in those cases that upper N-gram probabilities do not exist.

4 Word Graphs

Word graphs are successfully used in SMT for several applications. Basically, with the objective of reducing the redundancy of N-best lists, which very often convey serious combinatorial explosion problems.

A word graph is here described as a directed acyclic graph $G = (V, E)$ with one root node $n_0 \in V$. Edges are labeled with tokens (words or translation units) and optionally with accumulated scores. We will use $(n_s(n_e{''}t{''}\,s))$, to denote an edge starting at node n_s and ending at node n_e, with token t and score s. The file format of word graphs coincides with the graph file format recognized by the CARMEL[5] finite state automata toolkit.

4.1 Input Graph

We can mainly find two applications for which word graphs are used as input of an SMT system: the recognition output of an automatic speech recognition (ASR) system; and a reordering graph, consisting of a subset of the whole word permutations of a given input sentence.

In our case we are using the input graph as a **reordering graph**. The decoder introduces reordering (distortion of source words order) by allowing only for the distortion encoded in the input graph. Though, the graph is only allowed to encode permutations of the input words. In other words, any path in the graph must start at node n_0, finish at node n_N (where n_N is a unique ending node) and cover all the input words (tokens t) in whatever order, without repetitions.

An additional feature function (distortion model) is introduced in the log-linear combination of equation 1:

$$p_{distortion}(u_k) \approx \prod_{i=k_1}^{k_I} p(n_i|n_{i-1}) \tag{2}$$

where u_k refers to the k^{th} partial translation unit covering the source positions $[k_1, ..., k_I]$. $p(n_i|n_{i-1})$ corresponds to the edge score s encoded in the edge $(n_s(n_e{''}t{''}\,s))$, where $n_i = n_e$ and $n_{i-1} = n_s$.

One of the decoding first steps consists of building (for each input sentence) the set of translation units to be used in the search. When the search is extended with reordering abilities the set must be also extended with those translation units that cover any sequence of input words following any of the word orders encoded in the input graph. The extension of the units set is specially relevant when translation units are built from the tranining set with reordered source words.

Given the example of figure 2, if the translation unit '*translations perfect # traducciones perfectas*' is available, the decoder should not discard it, as it provides a right translation. Notwithstanding that its source side does not follow the original word order of the input sentence.

4.2 Output Graph

The goal of using an output graph is to allow for further rescoring work. That is, to work with alternative transla-

[5]http://www.isi.edu/licensed-sw/carmel/

tions to the single 1-best. Therefore, our proposed output graph has some peculiarities that make it different to the previously sketched intput graph.

The structure of edges remains the same, but obviously, paths are not forced to consist of permutations of the same tokens (as far as we are interested into multiple translation hypotheses), and there may also exist paths which do not reach the ending node n_N. These latter paths are not useful in rescoring tasks, but allowed in order to facilitate the study of the search graph. However, a very easy and efficient algorithm ($O(n)$, being n the search size) can be used in order to discard them, before rescoring work. Additionally, given that partial model costs are needed in rescoring work, our decoder allows to output the individual model costs computed for each translation unit (token t). Costs are encoded within the token s, as in the next example:

```
(0 (1 "o#or{1.5,0.9,0.6,0.2}" 6))
```

where the token t is now composed of the translation unit '*o#or*', followed by (four) model costs.

Multiple translation hypotheses can only be extracted if hypotheses recombinations are carefully saved. As in (Koehn, 2004), the decoder takes a record of any recombined hypothesis, allowing for a rigorous N-best generation. Model costs are referred to the current unit while the global score s is accumulated. Notice also that translation units (not words) are now used as tokens.

5 Experiments

Experiments are carried out for a Spanish-to-English translation task using the EPPS data set, corresponding to session transcriptions of the European Parliament.

Eff.	base	+tpos	+reor	+spos
Beam size $= 50$				
w/o cache	$1,820$	$2,170$	$2,970$	$3,260$
w/ cache	-50	-110	-190	-210
Beam size $= 100$				
w/o cache	$2,900$	$4,350$	$5,960$	$6,520$
w/ cache	-175	-410	-625	-640

Table 1: *Translation efficiency results.*

Table 1 shows translation efficiency results (measured in seconds) given two different beam search sizes. **w/cache** and **w/o cache** indicate whether the decoder employs (or not) the cache technique (section 3.3). Several system configuration have been tested: a baseline monotonous system using a 4-gram translation LM and a 5-gram target LM (**base**), extended with a target POS-tagged 5-gram LM (**+tpos**), further extended by allowing for reordering (**+reor**), and finally using a source-side POS-tagged 5-gram LM (**+spos**).

As it can be seen, the cache technique improves the efficiency of the search in terms of decoding time. Time results are further decreased (reduced time is shown for the **w/ cache** setting) by using more N-gram LM and allowing for a larger search graph (increasing the beam size and introducing distortion).

Further details on the previous experiment can be seen in (Crego and Mariño, 2006b; Crego and Mariño, 2006a), where additionally, the input word graph and extended N-gram tagged LM's are successfully used to improve accuracy at a very low computational cost.

Several publications can also be found in bibliography which show the use of output graphs in rescoring tasks allowing for clear accuracy improvements.

6 Conclusions

We have presented several extensions to MARIE, a freely available N-gram-based decoder. The extensions consist of accepting and generating word graphs, and introducing two N-gram LM's over source and target tagged words. Additionally, a caching technique is applied over the N-gram LM's.

Acknowledgments

This work has been funded by the European Union under the integrated project TC-STAR - (IST-2002-FP6-5067-38), the Spanish Government under the project AVIVAVOZ - (TEC2006-13694-C03) and the Universitat Politècnica de Catalunya under UPC-RECERCA grant.

References

J.M. Crego and J.B. Mariño. 2006a. Integration of postag-based source reordering into smt decoding by an extended search graph. *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas*, pages 29–36, August.

J.M. Crego and J.B. Mariño. 2006b. Reordering experiments for n-gram-based smt. *1st IEEE/ACL Workshop on Spoken Language Technology*, December.

J.M. Crego, J.B. Mariño, and A. de Gispert. 2005. An ngram-based statistical machine translation decoder. *Proc. of the 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 3193–3196, September.

Ph. Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Proc. of the 6th Conf. of the Association for Machine Translation in the Americas*, pages 115–124, October.

J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, and M.R. Costa-jussà. 2006. N-gram based machine translation. *Computational Linguistics*, 32(4):527–549.