Efficient Wordgraph Pruning for Interactive Translation Prediction

Germán Sanchis-Trilles
PRHLT Centre
Univ. Politéc. de Valencia
46022 Valencia, Spain
gsanchis@dsic.upv.es

Daniel Ortiz-Martínez
PRHLT Centre
Univ. Politéc. de Valencia
46022 Valencia, Spain
dortiz@dsic.upv.es

Francisco Casacuberta
PRHLT Centre
Univ. Politéc. de Valencia
46022 Valencia, Spain
fcn@dsic.upv.es

Abstract

When applying interactive translation prediction in real-life scenarios, response time is critical for the users to accept the interactive translation prediction system as a potentially useful tool. In this paper, we report on three different strategies for reducing the computation time required by a state-of-the-art interactive translation prediction system, so that automatic completions are delivered in real time. The best possibility turns out to be to directly prune the wordgraphs derived from the search procedure, achieving real-time response rates without any degradation whatsoever in the quality of the completions provided.

1 Introduction

Despite the recent advances in machine translation (MT) technology, MT systems are not able to provide ready-to-use translations in those contexts where translation accuracy is critical, such as medical or political applications, or even in contexts where correctness is demanded, such as hardware manuals or news texts. This has given rise to increasing research in computer assisted translation (CAT), where the focus is on how to provide a human translator with the best tools available in order to improve the human's efficiency. To this purpose, several ongoing FP7 projects were approved by the European Comission, some of them still being active. These projects pursue a very similar purpose, which is to develop a next generation CAT workbench.

One of the most innovative research directions regarding CAT tools implies interactive translation prediction (ITP) (Barrachina et al., 2009). Under this paradigm, system and human translator interact more closely than in a conventional post-editing setup, and the ITP engine attempts to provide improved completions for the sentence being translated after each one of the interactions of the human translator. Ideally, a constrained decoding, forced to produce the part of the sentence which has already been validated, should be performed before providing every suggestion. However, a full decoding process gives way to an important problem in ITP: the system needs to be able to provide translation completions in real time, since only a small delay in response time could easily lead users to reject the system. For this purpose, a common approximation is to extract a wordgraph off-line, i.e., before the user is actually sitting in front of the CAT tool. Then, during the ITP procedure, suggestions are obtained by searching for the best path in such a graph.

In the present work we report on different approaches analysed for the purpose of reducing the size of the wordgraph mentioned above when using a state-of-the-art ITP system. Since response time is critical, we studied three different strategies and measured the response time in a simulated ITP setup, alongside with an analysis of the degradation of the final translation quality obtained, both in terms of automatic MT metrics and in terms of simulated user effort.

The rest of this paper is structured as follows: in the next section, we briefly review the principles of ITP as an evolution of the classical SMT formulation. Then, in Section 4, we review the theoretical grounds of the strategies studied. Next, Section 5 reports the experiments conducted to assess the quality of the pruned wordgraphs and the response time associated. Finally, Section 6 presents the conclusions of the present work.

2 Statistical Framework

2.1 Statistical Machine Translation

Given a sentence $\mathbf{s}$ in a source language, the discipline of machine translation (MT) studies techniques to obtain its corresponding translation $\mathbf{t}$ in a target language by means of computer. Statistical MT (SMT) formalises this problem as follows (Brown et al., 1993):

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t} \mid \mathbf{s}) \tag{1}$$

$$= \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \tag{2}$$

The terms in the latter equation are the *language model* probability $\Pr(\mathbf{t})$ that represents the well-formedness of $\mathbf{t}$ (*n-gram* models are usually adopted), and the *translation model* $\Pr(\mathbf{s} \mid \mathbf{t})$ that represents the relationship between the source sentence and its translation.

In practice, all of these models (and possibly others) are often combined into a *log-linear model* for $\Pr(\mathbf{t} \mid \mathbf{s})$ (Och and Ney, 2002):

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \left\{ \sum_{n=1}^{N} \lambda_n \cdot \log(f_n(\mathbf{t}, \mathbf{s})) \right\} \tag{3}$$

where $f_n(\mathbf{t}, \mathbf{s})$ can be any model that represents an important feature for the translation, N is the number of models (or features), and λ_n are the weights of the log-linear combination.

One of the most popular instantiations of log-linear models is that including phrase-based models (Zens et al., 2002; Koehn et al., 2003) (Zens et al., 2002; Koehn et al., 2003). Phrase-based models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of phrase-based translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally to reorder the translated target phrases in order to compose the target sentence. Phrase-based models were employed throughout this work.

In log-linear models, the maximisation problem stated by Equation 3 is typically solved by means of dynamic programming-based algorithms (Zens et al., 2002), where the problem of translating a source sentence is decomposed into a set of partial solutions or hypotheses that are solved separately. A given partial hypothesis aligns a certain number of source words with words of the target language, and the rest remain unaligned. These hypotheses are stored in a stack (or priority queue) and ordered by their score. Such a score is given by the log-linear combination of feature functions.

source ($\mathbf{s}$): Para ver la lista de recursos
desired translation ($\hat{\mathbf{t}}$): To view a listing of resources

IT-0	$\mathbf{p}$	
	$\mathbf{t}_s$	*To view the resources list*
IT-1	$\mathbf{p}$	To view
	k	$\boxed{\mathbf{a}}$
	$\mathbf{t}_s$	*list of resources*
IT-2	$\mathbf{p}$	To view a list
	k	$\boxed{\mathbf{i}}$
	$\mathbf{t}_s$	*ng resources*
IT-3	$\mathbf{p}$	To view a listing
	k	$\boxed{\mathbf{o}}$
	$\mathbf{t}_s$	*f resources*
END	$\mathbf{p}$	To view a listing of resources

Figure 1: ITP session to translate a Spanish sentence into English. The desired translation is the translation the human user wants to obtain. At IT-0, the system suggests a translation ($\mathbf{t}_s$). At IT-1, the user moves the mouse to accept the first eight characters "To view " and presses the $\boxed{\text{a}}$ key (k), then the system suggests completing the sentence with "*list of resources*" (a new $\mathbf{t}_s$). Iterations 2 and 3 are similar. In the final iteration, the user accepts the current translation.

2.2 Interactive Translation Prediction

Unfortunately, current MT technology is not able to deliver error-free translations. This implies that, in order to achieve good translations, manual post-editing is needed. An alternative to this decoupled approach (first MT, then manual correction) is given by the ITP paradigm (Barrachina et al., 2009). Under this paradigm, translation is considered as an iterative left-to-right process where the human and the computer collaborate to generate the final translation.

Figure 1 shows an example of the ITP approach. There, a source Spanish sentence $\mathbf{s}$ ="Para ver la lista de recursos" is to be translated into a target English sentence $\hat{\mathbf{t}}$. Initially, with no user feedback, the system suggests a complete translation $\mathbf{t}_s$ ="To view the resources list". From this translation, the user marks a prefix $\mathbf{p}$ ="To view" as correct and begins to type the rest of the target sentence. Depending on the system or the user's preferences, the user might type the full next word, or only some letters of it (in our example, the user types the single next character "a"). Then, the MT system suggests a new suffix $\mathbf{t}_s$ ="list of resources" that completes the validated prefix and the input the user has just typed ($\mathbf{p}$ ="To view a"). The interaction continues with a new prefix validation followed, if necessary, by new input from the user, and so on, until the user considers the translation to be complete and satisfactory.

The crucial step of the process is the production of the suffix. Again, decision theory tells us to maximise the probability of the suffix given the available information. Formally, the best suffix of a given length will be:

$$\hat{\mathbf{t}}_s = \arg\max_{\mathbf{t}_s} \Pr(\mathbf{t}_s \mid \mathbf{s}, \mathbf{p}) \tag{4}$$

which can be straightforwardly rewritten as:

$$\begin{aligned} \hat{\mathbf{t}}_s &= \arg\max_{\mathbf{t}_s} \Pr(\mathbf{p}, \mathbf{t}_s \mid \mathbf{s}) & (5)\\ &= \arg\max_{\mathbf{t}_s} \Pr(\mathbf{p}, \mathbf{t}_s) \cdot \Pr(\mathbf{s} \mid \mathbf{p}, \mathbf{t}_s) & (6) \end{aligned}$$

Note that, since $\mathbf{p}\,\mathbf{t}_s = \mathbf{t}$, this equation is very similar to Equation (2). The main difference is that now the search process is restricted to those target sentences $\mathbf{t}$ that contain $\mathbf{p}$ as prefix. This implies that we can use the same MT models (including the log-linear approach) if the search procedures are adequately modified (Och et al., 2003a). Finally, it should be noted that the statistical models are usually defined at word level, while the ITP process described in this section works at character level. To deal with this problem, during the search process it is necessary to verify the compatibility between $\mathbf{t}$ and $\mathbf{p}$ at character level.

2.3 ITP with Stochastic Error-Correction

A common problem in ITP arises when the user sets a prefix which cannot be explained by the statistical models. To solve this problem, ITP systems typically include ad-hoc error-correction techniques to guarantee that the suffixes can be generated (Barrachina et al., 2009). As an alternative to this heuristic approach, Ortiz-Martínez (2011) proposed a formalisation of the ITP framework that does include stochastic error-correction models in its statistical formalisation. The starting point of this alternative ITP formalisation accounts for the problem of finding the translation $\mathbf{t}$ that, at the same time, better explains the source sentence $\mathbf{s}$ and the prefix given by the user $\mathbf{p}$:

$$\begin{aligned} \hat{\mathbf{t}} &= \arg\max_{\mathbf{t}} \Pr(\mathbf{t} \mid \mathbf{s}, \mathbf{p}) & (7)\\ &= \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s}, \mathbf{p} \mid \mathbf{t}) & (8) \end{aligned}$$

The following naïve Bayes' assumption is now made: the source sentence $\mathbf{s}$ and the user prefix $\mathbf{p}$ are statistically independent variables given the translation $\mathbf{t}$, obtaining:

$$\hat{\mathbf{t}} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{p} \mid \mathbf{t}) \tag{9}$$

where $\Pr(\mathbf{t})$ can be approximated with a language model, $\Pr(\mathbf{s} \mid \mathbf{t})$ can be approximated with a translation model, and $\Pr(\mathbf{p} \mid \mathbf{t})$ can be approximated by an error correction model that measures the compatibility between the user-defined prefix $\mathbf{p}$ and the hypothesized translation $\mathbf{t}$.

Note that the translation result, $\hat{\mathbf{t}}$, given by Equation (9) may not contain $\mathbf{p}$ as prefix because every translation is compatible with $\mathbf{p}$ with a certain probability. Thus, despite being close, Equation (9) is not equivalent to the ITP formalisation in Equation (6).

To solve this problem, we define an alignment, $\mathbf{a}$, between the user-defined prefix $\mathbf{p}$ and the hypothesised translation $\mathbf{t}$, so that the unaligned words of $\mathbf{t}$, in an appropriate order, constitute the suffix searched in ITP. This allows us to rewrite the error correction probability as follows:

$$\Pr(\mathbf{p} \mid \mathbf{t}) = \sum_{\mathbf{a}} \Pr(\mathbf{p}, \mathbf{a} \mid \mathbf{t}) \tag{10}$$

To simplify things, we assume that $\mathbf{p}$ is monotonically aligned to $\mathbf{t}$, leaving the potential word-reordering to the language and translation models. Under this assumption, $\mathbf{a}$ determines an alignment for $\mathbf{t}$, such that $\mathbf{t} = \mathbf{t}_p\mathbf{t}_s$, where $\mathbf{t}_p$ is fully-aligned to $\mathbf{p}$ and $\mathbf{t}_s$ remains unaligned. Taking all these things into consideration, and following a maximum approximation, we finally arrive to the expression:

$$(\hat{\mathbf{t}}, \hat{\mathbf{a}}) = \arg\max_{\mathbf{t},\mathbf{a}} \Pr(\mathbf{t}) \cdot \Pr(\mathbf{s} \mid \mathbf{t}) \cdot \Pr(\mathbf{p}, \mathbf{a} \mid \mathbf{t}) \tag{11}$$

where the suffix required in ITP is obtained as the portion of $\hat{\mathbf{t}}$ that is not aligned with the user prefix.

In practice, the models in Equation (11) are combined in a log-linear fashion as it is typically done in SMT (see Equation (3)).

2.4 ITP Using Wordgraphs

Common ITP implementations rely on a *wordgraph* data structure that represents possible translations of the given source sentence. A wordgraph is a weighted directed acyclic graph, in which each node represents a partial translation hypothesis and each edge is labelled with a word (or group of words) of the target sentence and is weighted according to the scores given by an SMT model (see (Ueffing et al., 2002) for more details). The use of wordgraphs in ITP has been studied in (Barrachina et al., 2009; Ortiz-Martínez, 2011; González-Rubio et al., 2013) in combination with different translation techniques.

The main advantage of wordgraph-based ITP systems is their efficiency in terms of the time cost per each interaction. This is due to the fact that the wordgraph is generated only once at the beginning of the interactive translation process of a given source sentence, and the suffixes required in ITP can be obtained by incrementally processing this wordgraph at each interaction.

All of the experiments reported in this paper always included stochastic error-correction for recovering from prefixes that cannot explained by the wordgraph for a given sentence.

3 Related Work

The use of wordgraphs in SMT was introduced in (Ueffing et al., 2002) for single word models and later extended to phrase-based models in (Zens and Ney, 2005). However, in these works, wordgraphs were applied within a fully-automatic SMT context. The first study on wordgraphs for ITP was given in (Och et al., 2003b). In that work, wordgraph pruning is used to speed-up suffix generation in an early ITP system based on the alignment template formalism. Bender et al. (Bender et al., 2005) extended the same strategy to a phrase-based ITP system with ad-hoc error correction techniques (see Section 2.3). Here, we propose efficient wordgraph pruning techniques for a state-of-the-art ITP system with stochastic error correction.

4 Efficient Suffix Generation in ITP

As it was explained in Section 2.4, common ITP formalisations, and more specifically, the one adopted in this paper, are typically based on the generation of wordgraphs.

The computational complexity of suffix generation using wordgraphs is linear in the number wordgraph states (Amengual and Vidal, 1998). Because of this, one possible way to achieve efficiency improvements would be to reduce the number of states per each wordgraph. One possible way to obtain smaller wordgraphs is to modify the pruning parameters that are applied during the decoding stage. Since wordgraphs constitute a compact representation of the search space explored by the SMT system, their size would be smaller if the search space is reduced as well. Another possibility to reduce the number of states contained in the wordgraph would be to apply pruning techniques directly over it.

4.1 Modifying Wordgraph Size in Decoding Time

To reduce the search space, regular SMT decoders based on dynamic programming have two different pruning parameters, namely, threshold pruning and histogram pruning (Och and Ney, 2002):

- **Threshold Pruning**: threshold pruning is applied for the different subsets of partial hypotheses that share the same number of already aligned source words. For a given subset, all those hypotheses whose score is below a certain percentage of the score of the best hypothesis for that subset are removed. The specific percentage used corresponds to the pruning threshold parameter.

- **Histogram Pruning**: the idea behind histogram pruning is to order those hypotheses that share the same number of already aligned source words by descending order of their scores, keeping only a certain quantity of the best of them.

4.2 Wordgraph Pruning

Threshold and histogram pruning constitute two possible techniques to reduce the wordgraph size during the decoding stage. Once the wordgraph has been generated, its size can be directly reduced using a technique that is closely related to threshold pruning. For this purpose, the probability of the best sentence hypothesis in the wordgraph is determined. After that, all those hypotheses in the graph whose probability is lower than this maximum probability multiplied by the pruning threshold are discarded. This wordgraph pruning technique was introduced in (Sixtus and Ortmanns, 1999) within the context of speech recognition.

The main difference between histogram and wordgraph pruning is that the former performs hypothesis pruning according to the score of the best partial hypothesis having a certain number of already aligned source words (i.e. pruning is locally applied) while the latter performs hypothesis pruning based on the probability of the best sentence hypothesis (a global pruning criterion is used).

If the wordgraph pruning threshold is zero, then the wordgraph is not pruned at all, and if the threshold is one, then only the sentence with maximum probability is retained.

5 Experimental Setup

In this section we detail the experimental setup designed to evaluate the different wordgraph size reduction strategies described in the previous section.

5.1 Corpora Used

The SMT system used to produce the translation models which later on were used to generate the wordgraphs were trained on the data provided for the ACL 2013 Workshop on Statistical Machine Translation (Bojar et al., 2013). Four training data sets were provided in this workshop: the Europarl corpus, the United Nations corpus, the Common Crawl corpus and the News Commentary corpus. Statistics of these data sets are provided in Table 1. As shown, these corpora together constitute a fair amount of data, and training an SMT system with all these data is computationally costly.

Additional development and test data were also considered (Table 2). For tuning the log-linear weights present in Equation 3, the test sets of the WMT 2008 to 2010 were considered, and the test set of WMT 2011 was considered as test data for the final evaluation.

5.2 System Description

For building the final ITP system, initial translation models were built by means of the open source Moses toolkit (Koehn et al., 2007)[1]. Then, the Moses decoder was also used for generating the wordgraphs. For doing this, the standard decoder configuration was used, i.e. a statistical log-linear model including a phrase-based translation model, a language model, a distortion model and word and phrase penalties. The baseline system was set up using the default threshold and histogram pruning parameters, i.e., 200 for the histogram pruning (200 maximum stack size) and 0.00001 for threshold pruning (hypothesis with a score less than 0.00001 times the best hypothesis are discarded). The weights of the log-linear combination are optimised by means of the Minimum Error Rate Training (MERT) procedure (Och, 2003).

		Es	En
Europarl	Sentences	1.9M	
	Run. words	54.0M	51.6M
	Vocabulary	181k	120.9k
United Nations	Sentences	10.8M	
	Run. words	317.6M	278.5M
	Vocabulary	612.0k	598.7k
Common Crawl	Sentences	1.8M	
	Run. words	46.7M	44.2M
	Vocabulary	763k	675.7k
News Com.	Sentences	172.8k	
	Run. words	5.0M	4.4M
	Vocabulary	88.8k	65.5k
Total	Sentences	14.7M	
	Run. words	423.3M	378.7M
	Vocabulary	1.2M	1.2M

Table 1: Statistics of the training data used in our experiments. These statistics are computed in tokenised and de-truecased conditions.

		Es	En
WMT08-10 Test	Sentences	7065	
	Run. words	186.2k	177.3k
	OoV words	1105	1073
WMT11 Test	Sentences	3003	
	Run. words	79.4k	74.8k
	OoV words	444	537

Table 2: Statistics of the WMT 2011 test data used to evaluate the system. These statistics are computed in tokenised and de-truecased conditions.

The phrase-based translation model provides direct and inverted frequency-based and lexical-based probabilities for each phrase pair in the phrase table. Phrase pairs are extracted from symmetrised word alignments generated by GIZA++ (Och and Ney, 2003). A 5-gram word-based LM is estimated on the target side of the parallel corpora using the improved Kneser-Ney smoothing (Chen and Goodman, 1999).

[1] Available from http://www.statmt.org/moses/

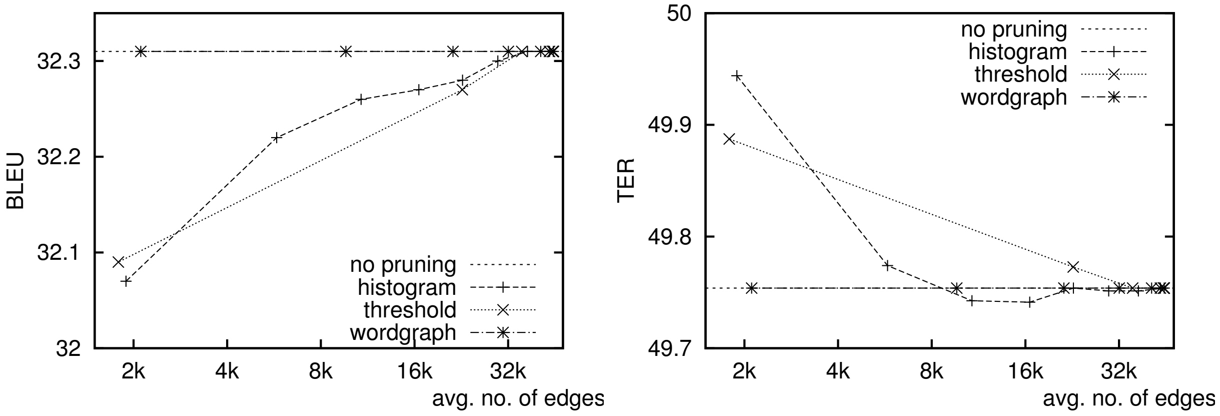

Figure 2: Translation quality of the best path in the reduced wordgraphs, measured both in terms of TER and BLEU. Note that the x-axis is in logarithmic scale for readability purposes.

For modelling word reordering, in addition to a negative-exponential on the reordering distance, a model conditioned on phrase pairs was estimated, namely the "orientation-bidirectional-fe" distortion model (Koehn et al., 2005).

Once the wordgraphs had been built, they were fed into the open-source Thot toolkit (Ortiz-Martínez and Casacuberta, 2014) [2], which implements among other things the ITP functionality used in this work. Such functionality allowed us to simulate real users by using the reference of the test data present in the corpus.

5.3 Assessment Metrics

The results produced by the ITP systems associated to the different wordgraph size reduction strategies were evaluated both in terms of conventional SMT metrics and ITP metrics. More specifically, the metrics used were:

- BLEU (Papineni et al., 2001) (Bilingual Language Evaluation Understudy) is an SMT precision metric that measures precision of unigrams, bigrams, trigrams and 4-grams, with a penalty for sentences that are too short.

- TER (Snover et al., 2006) (Translation Edit Rate) is an SMT error metric that computes the minimum number of edits required to modify the system hypotheses so that they match the references. Possible edits include insertion, deletion, substitution of single words and shifts of word sequences.

- KSMR (Barrachina et al., 2009) (Key Stroke Mouse-action Ratio) is an ITP error metric that measures the number of actions required by a human user to amend the system hypotheses so that they match the reference the user has in mind. Actions considered include key-strokes and the positioning of the mouse.

5.4 Results

We conducted experiments by testing different pruning thresholds according to the different strategies defined in Section 4. Figure 2 reports the final BLEU and TER scores achieved by the best hypothesis still present in the wordgraph after pruning has taken place. It is interesting to see that the wordgraph pruning strategy does not present any degradation as measured by TER and BLEU scores, while the other strategies do seem to correlate wordgraph size and translation quality. This is explained by the definition itself of wordgraph pruning strategy: since it only prunes the paths which fall beneath a given proportion of the probability of the best path, the best path itself is always preserved.

More interesting are the results of the ITP simulation, reported in Figure 3. Here it is shown that, just as in the case of BLEU and TER, KSMR seems to correlate quite evenly with wordgraph size in the case of histogram and threshold threshold strategies. However, when pruning the wordgraph directly, the human effort required to amend the hypotheses, as measured by KSMR, does not increase, and even presents a slight improvement for threshold values of 0.2 and 0.4 (equivalent to 21.3k and 9.6k edges on average, respectively). However, such improvement is not statistically significant and might be due to the effect of the stochastic error correction described in

[2] Available from https://github.com/daormar/thot/

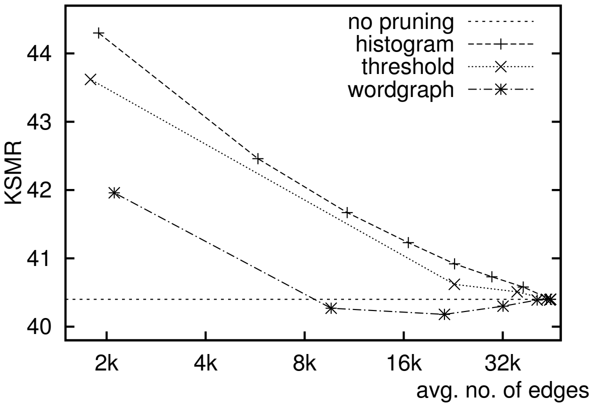

Figure 3: KSMR results when comparing different wordgraph sizes and the different pruning strategies described. Note that the x-axis is in logarithmic scale for readability purposes.

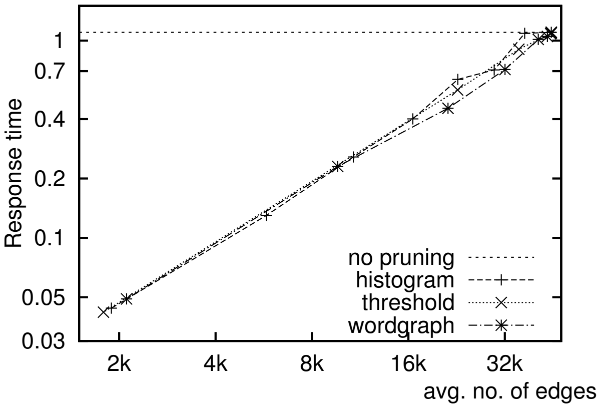

Figure 4: Average response time in seconds of the different systems when considering the different wordgraph size reduction strategies. Note that both the x-axis and the y-axis are in logarithmic scale.

Section 2.3. Nevertheless, it is important to point out that for these threshold values the amount of edges present in the wordgraph is reduced drastically, while no degradation in the performance of the system is observed until only around 8000 are left present in the wordgraph. The difference in behaviour between the BLEU and TER curves, on the one side, and the KSMR curves on the other side lies on the fact that KSMR is computed as an ITP simulation, and hence requires more information from the wordgraph than just its best path.

Finally, Figure 4 reports on the final response time required by the system. The experiments detailed here were performed on a multi-processor Intel Xeon E5-2650 @ 2.00GHz machine, with 64GBs memory, although each of the ITP simulations was not parallelised (i.e., each ITP simulation was executed sequentially). As shown, the complete wordgraph presents response times which are too high for a system set for online production. One could difficultly imagine that a potential user would wait for one second on average (much more in some cases) for the system to produce a hypothesis completion. However, by reducing the wordgraph by means of the wordgraph pruning strategy we are able to achieve real-time response times, while not having to compromise translation quality or human effort. Response time correlates empirically evenly with wordgraph size. When considering the 0.2 and 0.4 thresholds of the wordgraph pruning strategy, it was observed that the average response times were of 0.23 and 0.05 seconds, respectively, which is perfectly suitable for an ITP system set for online production. Moreover, it must be emphasised that such speed increase is achieved without any degradation of system performance measured in terms of KSMR.

6 Conclusions

In this paper we have compared three approaches for obtaining smaller-sized wordgraphs for the purpose of providing sentence completions by means of a state-of-the-art ITP engine. We have seen that regular wordgraphs, as produced by a state-of-the-art decoder, imply too much computational time for their usage within an ITP system. We have also shown that pruning the wordgraph directly by removing those paths whose probability falls below a certain proportion of the probability of the best path is able to yield completions with exactly the same quality as the un-pruned wordgraphs, but with much better response times.

We understand that the analysis performed in this work is crucial for research in ITP, since hypothesis completion times above one second can be considered unacceptable for a human translator. The pruning techniques proposed in this paper allow us to solve this issue effectively.

Acknowledgements

Work supported by the European Union 7$^{\text{th}}$ Framework Program (FP7/2007-2013) under the CasMaCat project (grant agreement n$^{\text{o}}$ 287576), by Spanish MICINN under grant TIN2012-31723, and by the Generalitat Valenciana under grant ALMPR (Prometeo/2009/014).

References

Amengual, J.C. and E. Vidal. 1998. Efficient error-correcting viterbi parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-20, No.10:1109–1116, October.

Barrachina, Sergio, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35:3–28, March.

Bender, O., S. Hasan, D. Vilar, R. Zens, and H. Ney. 2005. Comparison of generation strategies for interactive machine translation. In *Conference of the European Association for Machine Translation*, pages 33–40, Budapest, Hungary, May.

Bojar, Ondrej, Christian Buck, Chris Callison-Burch, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Herve Saint-Amand, Radu Soricut, and Lucia Specia, editors. 2013. *Proceedings of the Eighth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Sofia, Bulgaria, August.

Brown, Peter F., Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311.

Chen, Stanley F. and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393.

González-Rubio, Jesús, Daniel Ortiz-Martínez, José-Miguel Benedí, and Francisco Casacuberta. 2013. Interactive machine translation using hierarchical translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 244–254.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, pages 48–54, Edmonton, Canada, May.

Koehn, P., A. Axelrod, A. Birch Mayne, C. Callison-Burch, M. Osborne, and D. Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proc. of IWSLT*, Pittsburgh, PA.

Koehn et al., P. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Och, Franz Josef and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302.

Och, F. J. and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz Josef, Richard Zens, and Hermann Ney. 2003a. Efficient search for interactive statistical machine translation. In *Proceedings of the European chapter of the Association for Computational Linguistics*, pages 387–393.

Och, Franz Josef, Richard Zens, and Hermann Ney. 2003b. Efficient search for interactive statistical machine translation. In *In EACL 03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 387–393.

Och, F.J. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*, pages 160–167, Sapporo, Japan.

Ortiz-Martínez, D. and F. Casacuberta. 2014. The new Thot toolkit for fully automatic and interactive statistical machine translation. In *14th Annual Meeting of the European Association for Computational Linguistics: System Demonstrations*, pages 45–48, Gothenburg, Sweden, April.

Ortiz-Martínez, Daniel. 2011. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation*. Ph.D. thesis, Universitat Politècnica de València. Advisors: Ismael García Varea and Francisco Casacuberta.

Papineni, K., A. Kishore, S. Roukos, T. Ward, and W. Jing Zhu. 2001. Bleu: A method for automatic evaluation of machine translation. In *Technical Report RC22176 (W0109-022)*.

Sixtus, Achim and Stefan Ortmanns. 1999. High quality word graphs using forward-backward pruning. In *Proceedins of the ICASSP*, pages 593–596.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proc. of AMTA'06*.

Ueffing, N., F. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. pages 156–163.

Zens, Richard and Hermann Ney. 2005. Word graphs for statistical machine translation. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 191–198, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Zens, R., F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Advances in artificial intelligence. 25. Annual German Conference on AI*, volume 2479 of *LNCS*, pages 18–32. Springer Verlag, September.