SYNTACTIC PHRASE-BASED STATISTICAL MACHINE TRANSLATION

Hany Hassan, Mary Hearne, Andy Way[*]

School of Computing,
Dublin City University,
Dublin 9, Ireland
{hhasan,mhearne,away}@computing.dcu.ie

Khalil Sima'an

ILLC,
University of Amsterdam,
1018 TV Amsterdam, The Netherlands
simaan@science.uva.nl

ABSTRACT

Phrase-based Statistical Machine Translation (PBSMT) systems represent the dominant approach in MT today. However, unlike systems in other paradigms, it has proven difficult to date to incorporate syntactic knowledge in order to improve translation quality. This paper improves on recent research which uses 'syntactified' target language phrases, by incorporating supertags as constraints to better resolve parse tree fragments. In addition, we do not impose any sentence-length limit, and using a log-linear decoder, we outperform a state-of-the-art PBSMT system by over 1.3 BLEU points (or 3.51% relative) on the NIST 2003 Arabic–English test corpus.

1. INTRODUCTION

Almost all research in MT being carried out today is corpus-based. Within this field, by far the most dominant paradigm is Phrase-based Statistical Machine Translation (PBSMT) [6, 9, 12, 17]. However, unlike in rule- and example-based MT, it has proven difficult to date to incorporate syntactic knowledge in order to improve translation quality. For example, [6] actually demonstrated that adding syntax harmed the quality of their SMT system.

More recently, [2] demonstrates significant improvements over the baseline by allowing for hierarchical phrase probabilities to handle a range of linguistic phenomena in the correct fashion. However, the derived grammar does not rely on any linguistic annotations or assumptions, so that the 'syntax' induced is not linguistically motivated.

Coming right up to date, [8] demonstrate that 'syntactified' target language phrases can improve translation quality for Chinese–English. While this research has much in common with the approach proposed here, there remain a number of significant differences: (i) rather than induce millions of xRS rules from parallel data, we extract phrase pairs in the usual manner [11] and associate with each phrase-pair a set of target-language syntactic constituents based on supertags [1]; (ii) unlike [8], who restrict their experiments to sentences of max. 20 words, we do not impose any such sentence-length limit; (iii) instead of using a CKY-style decoder, we deploy a log-linear, left-to-right decoder [5, 10]; (iv) unlike [8], we have no need to resort to *ad hoc* tree-rewriting measures in order to provide a better interaction between 'good' ('normal' PBSMT) and 'bad' (xRS) rules.

The remainder of the paper is organised as follows: in section 2, we detail our approach. Section 3 describes the experiments carried out, together with the results obtained. Section 4 concludes, and provides avenues for further work.

2. OUR APPROACH

As in any state-of-the-art PBSMT system, our approach extracts phrase correspondences from a bilingual training corpus. We use the method of [11] to extract phrase correspondences from bidirectional symmetrized word alignments obtained via Giza++. While in principle any arbitrary (monolingual source or target language, or bilingual) constraints can be used in our approach, for the purposes of the experiments described here, we use supertags [1] to decorate the boundaries of these phrases to derive a set of target-language syntactic constituents. As part of the translation process, we use a log-linear decoder with an added cost function for these constraints along with the 'normal' translation and language models.

2.1. The Model

In this section we describe the model used more formally. Let t and s be the target and source language sentences. Any (target or source) sentence x will consist of two parts, a bag of elements (words/phrases etc.) and an order over that bag; in other words, $x = \langle \{x\}, O_x \rangle$, where $\{x\}$ stands for the bag of word tokens (or phrases) that constitute x, and O_x for the order of the word tokens (respectively phrases) as given in x (O_x can be implemented as a function from a bag of tokens $\{x\}$ to a set with a finite number of positions).

$$\begin{aligned} \arg\max_t P(t|s) &= \arg\max_{\langle \{t\}, O_t \rangle} P(\{s\}, O_s \mid \{t\}, O_t) P(\{t\}, O_t) \\ &= \arg\max_{\langle \{t\}, O_t \rangle} P(\{s\} \mid \{t\})\ P(O_s \mid O_t)\ P(t) \end{aligned}$$

[*]Thanks to Science Foundation Ireland (http://www.sfi.ie) Principal Investigator Award 05/IN/1732 for part-funding this research.

$P(t) = P(\{t\}, O_t)$ stands for the target language model, $P(O_s|O_t)$ represents the conditional distortion probability, and $P(\{s\}|\{t\})$ stands for a probabilistic translation model from target language bags of phrases to source language bags of phrases.

2.2. Language Modeling Using Supertags

Usually a language model over a finite vocabulary V assigns a probability to every finite sequence of words in the formal language V^+, i.e. a language model implements a function $P : V^+ \rightarrow [0, 1]$ such that $\sum_{x \in V^+} P(x) = 1$. Language models implemented using Markov models over bare word sequences are common in both speech recognition and MT. The statistics of such Markov models are based on frequencies of n-grams. While language models over bare word sequences are robust and useful, they are not suitable for grading sentences on their grammatical well-formedness.

Naturally, language models may also incorporate grammatical structure. The problem usually is how much grammatical structure can be incorporated without resulting in sparse statistics (and thereby losing the required robustness). Within Markov-based language models, the (impoverished) grammatical structure is usually assumed to consist of a finite set of word categories. Let C stand for the finite set of word categories for words in the vocabulary V. A language model can be obtained from the joint probabilities $P(x, y)$, for sequences $x \in V^n$ and $y \in C^n$ (for all $n \leq 1$) through the formula $P(x) = \sum_{y \in C^n} P(x, y)$ (thereby assuming that the occurrences of word categories are mutually exclusive). Hence, incorporating the word categories results in Hidden Markov Models (HMMs) [15].

A popular linguistic approach for defining word categories (the set C) is via part-of-speech (POS) tag categories. POS taggers based on HMMs are well-known in the literature [7]. An HMM POS tagger assigns a joint probability $P(x, y)$ to a sentence $x \in V^n$ and to a POS tag sequence $y \in C^n$, through the two-step generative process $P(x, y) = P(y)P(x|y)$, whereby $P(y)$ is a Markov language model over POS tag sequences and $P(x|y)$ is the lexical model, where probabilities of words are conditioned on POS tags. However, the commonly used POS tags impose only a very weak set of grammatical constraints (depending on the kind of POS tags employed of course). In [1], a more advanced linguistic alternative is proposed: Supertags.

A supertag stands for a complex, linguistic word category that encodes a syntactic structure that unambiguously expresses a specific local behaviour of a word, in terms of the arguments (e.g., subject, object) it takes and the syntactic environment in which it appears. A sequence of supertags specifies "almost a parse" [1] in the sense that if the ordered sequences of supertags combine together under the combinatory operators (substitution and adjunction) of Tree-Adjoining Grammar (TAG) [4], only a little extra effort is required in order to obtain the full parse tree that these supertags specify.

While the original TAG that underlies the supertagging approach is not directly employed within supertagging, the conceptual way in which TAG describes language is crucial for understanding what supertags are. In the original TAG grammatical framework, the supertags are elementary trees (grammar productions). A TAG consists of two disjoint finite sets of elementary trees, initial and auxiliary. The initial elementary trees combine through the well-known substitution operation (as in Context-Free Grammars) and result in initial sentences/trees that do not contain recursion (so called adjuncts, e.g. prepositional phrases and adjectival phrases). The auxiliary elementary trees are recursive tree structures that combine through the special adjunction operator and lead to sentences/trees that contain recursion in the form of adjuncts. Hence, a Lexicalized TAG system consists of a lexicon (the initial and auxiliary elementary trees, each lexicalized with a word from the language) and the two combinatory TAG operators, substitution and adjunction. In the TAG system for English, the lexicon consists of about 5000 supertags (unlexicalized elementary trees).

The supertagger of [1] is a standard HMM tagger and consists of a (second-order) Markov language model over supertags and a lexical model conditioning the probability of every word on its own supertag (just like standard HMM-based POS taggers). In this work, this supertagger is employed as a language model, i.e. it assigns a probability to every input word sequence as discussed above. Next we explain the details of how this supertag-based language model is integrated into the decoder.

2.3. A Decoder Using Supertags

Our decoder is a log-linear decoder similar to Pharaoh [5], with the main modification being the addition of a constraint-based target language model. The decoder is built on the MOOD framework described in [14]. During decoding three feature costs are computed: the phrase translation probability, the ('regular' trigram, backing off to lower orders) target language model probability and the supertag, constraint-based target language model probability (5-gram, backing off to lower orders).

3. EXPERIMENTS

3.1. Resources

We translated from Arabic to English, training the system on 180K sentences (5M words) of the Arabic–English news parallel corpus from the LDC. The n-gram target language model was built using 250M words from the English GigaWord Cor-

pus using the SRI language modelling toolkit [16].[1] Taking 10% of the English GigaWord Corpus used for building our target language model, the supertag constraint-based target language model was built from 25M supertags obtained via the XTAG English supertagger.[2]

3.2. Baseline vs. Extended System

The baseline system that we use to compare our model is exactly the same as the extended model minus the constraint-based target language model. As an example, the baseline system is trained on source–target phrases such as that in (1):

(1) $Anh\ ATlE \Leftrightarrow that\ he\ brie fed$

For the improved model proposed here, the English phrase is supertagged as in (2):

(2) $that$ `//IN-B-COMPs` he `//PRP//A-NXG`
$brie fed$ `//VBD//A-nx0Vnx1`

The resultant constraints, therefore, would be those in (3):

(3) `IN//B-COMPs PRP//A-NXG`
`VBD//A-nx0Vnx1.`

3.3. Results

System	BLEU Score
Baseline-60K	.3756
S-PBSMT-60K	.3888
Baseline-180K	.4088
S-PBSMT-180K	.4194

Table 1. Comparing the Baseline and S-PBSMT Systems: the first pair of systems are trained on 60K sentence pairs, while the second pair of systems are trained on 180K sentence pairs.

The results are given in Table 1. Training the baseline and extended Syntactic PBSMT (S-PBSMT) systems on 60K sentence pairs, and testing on 663 Arabic sentences (ave. 25 words, min. 4 words, max. 71 words)—the standard NIST 2003 evaluation test set—the S-PBSMT system obtains a BLEU score [13] of 0.3888, 1.32 points (or 3.51% relative) better than the baseline. When testing on 180K sentence pairs, and testing on the same test set, the S-PBSMT system scores 0.4194 for BLEU, 1.06 points (or 2.59% relative) better than the baseline.[3] Some sample output is given in Figure 1.

[1] `http://www.speech.sri.com/projects/srilm/`

[2] `http://www.cis.upenn.edu/~xtag/gramrelease.html`

[3] We preprocessed all numbers so that they were classed as a single category. Any OOV items which occurred in the test set were replaced by their nearest Arabic form if in the vocabulary. All scores shown are for lowercased English. Four reference translations were provided.

$Reference$: Saudi sources this week denied reports in the American New York Times that Saudi Arabia had agreed to allow the United States to use Saudi military bases should a war against Iraq take place.

$Baseline$: the saudis denied this week in new york times , saudi arabia agreed to lay down its military stock in the united states in the event of war with iraq. american information published

$S-PBSMT$: the saudis denied information published this week in new york times , saudi arabia agreed to lay down its military in the united states , in the event of war with iraq . american

$Reference$: He added: "The position of the Kingdom regarding this matter has been clear from the beginning, and we are not able to place our air space .....

$Baseline$: he added that " saudi arabia clear of the this we cannot be put our airspace .....

$S-PBSMT$: he added that " the attitude of the kingdom is clear this we cannot be put our airspace ...

Fig. 1. Sample Output

4. CONCLUSIONS AND FUTURE WORK

Despite being the dominant approach in MT today, developers of PBSMT systems have found it difficult to integrate syntactic knowledge with a resultant increase in translation quality. [8] have recently demonstrated that 'syntactified' target language phrases can improve translation quality for Chinese–English.

In this paper, we have improved on the method of [8] by imposing no sentence-length limit, and employing a log-linear, left-to-right decoder instead of a CKY-style decoder. In a series of experiments, we have shown how syntactic constraints in the form of supertags can improve translation quality for Arabic–English by between 2.6% and 3.5% relative BLEU score.

Furthermore, while only using target language constraints here, our architecture allows for any constraints to be integrated into a PBSMT system. In future work, we intend to factor in source language constraints, as well as bilingual constraints induced via the DOT alignments of [3]. Other constraints could be included via by leaf path projection [18]; note that both these approaches allow for long-distance dependencies to be handled successfully. Finally, we hope to investigate more sophisticated scoring methods for handling the constraints, such as 'almost parsing' techniques using finite-state machines.

5. REFERENCES

[1] S. Bangalore and A. Joshi, "Supertagging: An Approach to Almost Parsing", *Computational Linguistics*

25(2):237–265, 1999.

[2] D. Chiang, "A Hierarchical Phrase-Based Model for Statistical Machine Translation", in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, MI., pp.263–270, 2005.

[3] D. Groves, M. Hearne, and A. Way, "Robust Sub-Sentential Alignment of Phrase-Structure Trees", in *Proceedings of The 20th International Conference on Computational Linguistics (COLING'04)*, Geneva, Switzerland, pp.1072–1078, 2004.

[4] A. Joshi and Y. Schabes, "Tree Adjoining Grammars and Lexicalized Grammars" in M. Nivat and A. Podelski (eds.) *Tree Automata and Languages*, Amsterdam, The Netherlands: North-Holland, pp.409–431, 1992.

[5] P. Koehn, "Pharaoh: A Beam Search Decoder for phrase-based Statistical Machine Translation Models", in R. Frederking & K. Taylor (eds.) *Machine Translation: From Real Users to Research; 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004*, LNAI 3265, Berlin/Heidelberg, Germany: Springer Verlag, pp.115–124, 2004.

[6] P. Koehn, F. Och, and D. Marcu, "Statistical Phrase-Based Translation", in *Proceedings of the Joint Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Edmonton, Canada, pp.127–133, 2003.

[7] J. Kupiec, "Robust Part-of-Speech tagging Using a Hidden Markov Model", *Computer Speech and Language* **6**:225–242, 1992.

[8] D. Marcu, W. Wang, A. Echihabi and K. Knight, "SPMT: Statistical Machine Translation with Syntactified Target language Phrases", in *Proceedings of EMNLP*, Sydney, Australia, 2006.

[9] D. Marcu and W. Wong, "A Phrase-Based, Joint Probability Model for Statistical Machine Translation", in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-02)*, Philadelphia, PA., pp.133–139, 2002.

[10] F. Och and H. Ney, "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation", in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA., pp.295–302, 2002.

[11] F. Och and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics* **29**:19–51, 2003.

[12] F. Och, C. Tillmann, and H. Ney, "Improved Alignment Models for Statistical Machine Translation", in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, College Park, MD., pp.20–28, 1999.

[13] K. Papineni, S. Roukos, T. Ward and W-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation", in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA., pp.311–318, 2002.

[14] A. Patry, F. Gotti, and P. Langlais, "MOOD: A Modular Object-Oriented Decoder for Statistical Machine Translation", in *Proceedings of 5th LREC*, Genoa, Italy, pp.709–714, 2006.

[15] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", in A. Waibel & F-K. Lee (eds.) *Readings in Speech Recognition*, San Mateo, CA.: Morgan Kauffmann, pp.267–296, 1990.

[16] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit", in *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, CO, pp.901–904, 2002.

[17] C. Tillmann and F. Xia, "A Phrase-based Unigram Model for Statistical Machine Translation", in *Proceedings of HLT-NAACL 2003*, Edmonton, Canada. pp.106–108, 2003.

[18] K. Toutanova, P. Markova, and C. Manning, "The leaf projection path view of parse trees: Exploring string kernels for HPSG parse selection", in *Proceedings of EMNLP 2004*, Barcelona, Spain, pp.166–173, 2004.