Supertagged Phrase-Based Statistical Machine Translation

Hany Hassan
School of Computing,
Dublin City University,
Dublin 9, Ireland
hhasan@computing.dcu.ie

Khalil Sima'an
Language and Computation,
University of Amsterdam,
Amsterdam, The Netherlands
simaan@science.uva.nl

Andy Way
School of Computing,
Dublin City University,
Dublin 9, Ireland
away@computing.dcu.ie

Abstract

Until quite recently, extending Phrase-based Statistical Machine Translation (PBSMT) with syntactic structure caused system performance to deteriorate. In this work we show that incorporating lexical syntactic descriptions in the form of supertags can yield significantly better PBSMT systems. We describe a novel PBSMT model that integrates supertags into the target language model and the target side of the translation model. Two kinds of supertags are employed: those from Lexicalized Tree-Adjoining Grammar and Combinatory Categorial Grammar. Despite the differences between these two approaches, the supertaggers give similar improvements. In addition to supertagging, we also explore the utility of a surface global grammaticality measure based on combinatory operators. We perform various experiments on the Arabic to English NIST 2005 test set addressing issues such as sparseness, scalability and the utility of system subcomponents. Our best result (0.4688 BLEU) improves by 6.1% relative to a state-of-the-art PBSMT model, which compares very favourably with the leading systems on the NIST 2005 task.

1 Introduction

Within the field of Machine Translation, by far the most dominant paradigm is Phrase-based Statistical Machine Translation (PBSMT) (Koehn et al., 2003; Tillmann & Xia, 2003). However, unlike in rule- and example-based MT, it has proven difficult to date to incorporate linguistic, syntactic knowledge in order to improve translation quality. Only quite recently have (Chiang, 2005) and (Marcu et al., 2006) shown that incorporating some form of syntactic structure could show improvements over a baseline PBSMT system. While (Chiang, 2005) avails of structure which is not linguistically motivated, (Marcu et al., 2006) employ syntactic structure to enrich the entries in the phrase table.

In this paper we explore a novel approach towards extending a standard PBSMT system with syntactic descriptions: we inject *lexical* descriptions into both the target side of the phrase translation table and the target language model. Crucially, the kind of lexical descriptions that we employ are those that are commonly devised within lexicon-driven approaches to linguistic syntax, e.g. Lexicalized Tree-Adjoining Grammar (Joshi & Schabes, 1992; Bangalore & Joshi, 1999) and Combinary Categorial Grammar (Steedman, 2000). In these linguistic approaches, it is assumed that the grammar consists of a very rich lexicon and a tiny, *impoverished*[1] set of combinatory operators that assemble lexical entries together into parse-trees. The lexical entries consist of syntactic constructs ('supertags') that describe information such as the POS tag of the word, its subcategorization information and the hierarchy of phrase categories that the word projects upwards. In this work we employ the lexical entries but exchange the algebraic combinatory operators with the more robust

[1] These operators neither carry nor presuppose further linguistic knowledge beyond what the lexicon contains.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 288–295,
Prague, Czech Republic, June 2007.

and efficient *supertagging* approach: like standard taggers, supertaggers employ probabilities based on local context and can be implemented using finite state technology, e.g. Hidden Markov Models (Bangalore & Joshi, 1999).

There are currently two supertagging approaches available: LTAG-based (Bangalore & Joshi, 1999) and CCG-based (Clark & Curran, 2004). Both the LTAG (Chen et al., 2006) and the CCG supertag sets (Hockenmaier, 2003) were acquired from the WSJ section of the Penn-II Treebank using hand-built extraction rules. Here we test both the LTAG and CCG supertaggers. We interpolate (log-linearly) the supertagged components (language model and phrase table) with the components of a standard PBSMT system. Our experiments on the Arabic–English NIST 2005 test suite show that each of the supertagged systems significantly improves over the baseline PBSMT system. Interestingly, combining the two taggers together diminishes the benefits of supertagging seen with the individual LTAG and CCG systems. In this paper we discuss these and other empirical issues.

The remainder of the paper is organised as follows: in section 2 we discuss the related work on enriching PBSMT with syntactic structure. In section 3, we describe the baseline PBSMT system which our work extends. In section 4, we detail our approach. Section 5 describes the experiments carried out, together with the results obtained. Section 6 concludes, and provides avenues for further work.

2 Related Work

Until very recently, the experience with adding syntax to PBSMT systems was negative. For example, (Koehn et al., 2003) demonstrated that adding syntax actually harmed the quality of their SMT system. Among the first to demonstrate improvement when adding recursive structure was (Chiang, 2005), who allows for hierarchical phrase probabilities that handle a range of reordering phenomena in the correct fashion. Chiang's 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, (Marcu et al., 2006) demonstrate that 'syntactified' target language phrases can improve translation quality for Chinese–English. They employ a stochastic, top-down transduction process that assigns a joint probability to a source sentence and each of its alternative translations when rewriting the target parse-tree into a source sentence. The rewriting/transduction process is driven by "xRS rules", each consisting of a pair of a source phrase and a (possibly only partially) lexicalized syntactified target phrase. In order to extract xRS rules, the word-to-word alignment induced from the parallel training corpus is used to guide heuristic tree 'cutting' criteria.

While the research of (Marcu et al., 2006) has much in common with the approach proposed here (such as the syntactified target phrases), there remain a number of significant differences. Firstly, rather than induce millions of xRS rules from parallel data, we extract phrase pairs in the standard way (Och & Ney, 2003) and associate with each phrase-pair a set of target language syntactic structures based on supertag sequences. Relative to using arbitrary parse-chunks, the power of supertags lies in the fact that they are, syntactically speaking, rich lexical descriptions. A supertag can be assigned to every word in a phrase. On the one hand, the correct sequence of supertags could be assembled together, using only impoverished combinatory operators, into a small set of constituents/parses ('almost' a parse). On the other hand, because supertags are lexical entries, they facilitate robust syntactic processing (using Markov models, for instance) which does not necessarily aim at building a fully connected graph.

A second major difference with xRS rules is that our supertag-enriched target phrases *need not* be generalized into (xRS or any other) rules that work with abstract categories. Finally, like POS tagging, supertagging is more efficient than actual parsing or tree transduction.

3 Baseline Phrase-Based SMT System

We present the baseline PBSMT model which we extend with supertags in the next section. Our baseline PBSMT model uses GIZA++[2] to obtain word-level alignments in both language directions. The bidirectional word alignment is used to obtain phrase translation pairs using heuristics presented in

[2]http://www.fjoch.com/GIZA++.html

(Och & Ney, 2003) and (Koehn et al., 2003), and the Moses decoder was used for phrase extraction and decoding.[3]

Let t and s be the target and source language sentences respectively. 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, $x = \langle \phi_x, O_x \rangle$, where ϕ_x stands for the bag of phrases that constitute x, and O_x for the order of the phrases as given in x (O_x can be implemented as a function from a bag of tokens ϕ_x to a set with a finite number of positions). Hence, we may separate order from content:

$$\arg\max_t P(t|s) = \arg\max_t P(s \mid t)P(t) \qquad (1)$$

$$= \arg\max_{\langle \phi_t, O_t \rangle} \overbrace{P(\phi_s \mid \phi_t)}^{TM}\ \overbrace{P(O_s \mid O_t)}^{distortion}\ \overbrace{P_w(t)}^{LM} \qquad (2)$$

Here, $P_w(t)$ is the target language model, $P(O_s|O_t)$ represents the conditional (order) linear distortion probability, and $P(\phi_s|\phi_t)$ stands for a probabilistic translation model from target language bags of phrases to source language bags of phrases using a phrase translation table. As commonly done in PBSMT, we interpolate these models log-linearly (using different λ weights) together with a word penalty weight which allows for control over the length of the target sentence t:

$$\arg\max_{\langle \phi_t, O_t \rangle} P(\phi_s \mid \phi_t)\ P(O_s \mid O_t)^{\lambda_o}$$
$$P_w(t)^{\lambda_{lm}}\ exp^{|t|\lambda_w}$$

For convenience of notation, the interpolation factor for the bag of phrases translation model is shown in formula (3) at the phrase level (but that does not entail any difference). For a bag of phrases ϕ_t consisting of phrases t_i, and bag ϕ_s consisting of phrases s_i, the phrase translation model is given by:

$$P(\phi_s \mid \phi_t) = \prod_{\substack{s_i \\ t_i}} P(s_i|t_i)$$

$$P(s_i|\ t_i) = P_{ph}(s_i|t_i)^{\lambda_{t1}} P_w(s_i|t_i)^{\lambda_{t2}} P_r(t_i|s_i)^{\lambda_{t3}} \qquad (3)$$

where P_{ph} and P_r are the phrase-translation probability and its reverse probability, and P_w is the lexical translation probability.

[3] http://www.statmt.org/moses/

4 Our Approach: Supertagged PBSMT

We extend the baseline model with lexical linguistic representations (*supertags*) both in the language model as well as in the phrase translation model. Before we describe how our model extends the baseline, we shortly review the supertagging approaches in Lexicalized Tree-Adjoining Grammar and Combinatory Categorial Grammar.

4.1 Supertags: Lexical Syntax

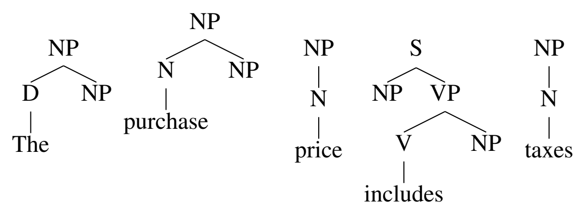

Figure 1: An LTAG supertag sequence for the sentence *The purchase price includes taxes*. The subcategorization information is most clearly available in the verb *includes* which takes a subject NP to its left and an object NP to its right.

Modern linguistic theory proposes that a syntactic parser has access to an extensive lexicon of word-structure pairs and a small, impoverished set of operations to manipulate and combine the lexical entries into parses. Examples of formal instantiations of this idea include CCG and LTAG. The lexical entries are syntactic constructs (graphs) that specify information such as POS tag, subcategorization/dependency information and other syntactic constraints at the level of agreement features. One important way of portraying such lexical descriptions is via the supertags devised in the LTAG and CCG frameworks (Bangalore & Joshi, 1999; Clark & Curran, 2004).

A supertag (see Figure 1) represents a complex, linguistic word category that encodes a syntactic structure expressing a specific local behaviour of a word, in terms of the arguments it takes (e.g. subject, object) and the syntactic environment in which it appears. In fact, in LTAG a supertag is an elementary tree and in CCG it is a CCG lexical category. Both descriptions can be viewed as closely related functional descriptions.

The term "supertagging" (Bangalore & Joshi, 1999) refers to tagging the words of a sentence, each with a supertag. When well-formed, an ordered sequence of supertags can be viewed as a compact representation of a small set of constituents/parses that can be obtained by assembling the supertags together using the appropriate combinatory operators (such as substitution and adjunction in LTAG or function application and combination in CCG). Akin to POS tagging, the process of *supertagging* an input utterance proceeds with statistics that are based on the probability of a word-supertag pair given their Markovian or local context (Bangalore & Joshi, 1999; Clark & Curran, 2004). This is the main difference with full parsing: supertagging the input utterance need not result in a fully connected graph.

The LTAG-based supertagger of (Bangalore & Joshi, 1999) 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).

The CCG supertagger (Clark & Curran, 2004) is based on log-linear probabilities that condition a supertag on features representing its context. The CCG supertagger does not constitute a language model nor are the Maximum Entropy estimates directly interpretable as such. In our model we employ the CCG supertagger to obtain the best sequences of supertags for a corpus of sentences from which we obtain language model statistics. Besides the difference in probabilities and statistical estimates, these two supertaggers differ in the way the supertags are extracted from the Penn Treebank, cf. (Hockenmaier, 2003; Chen et al., 2006). Both supertaggers achieve a supertagging accuracy of 90–92%.

Three aspects make supertags attractive in the context of SMT. Firstly, supertags are rich syntactic constructs that exist for individual words and so they are easy to integrate into SMT models that can be based on any level of granularity, be it word- or phrase-based. Secondly, supertags specify the local syntactic constraints for a word, which resonates well with sequential (finite state) statistical (e.g. Markov) models. Finally, because supertags are rich lexical descriptions that represent underspecification in parsing, it is possible to have some of the benefits of full parsing without imposing the strict connectedness requirements that it demands.

4.2 A Supertag-Based SMT model

We employ the aforementioned supertaggers to enrich the English side of the parallel training corpus with a single supertag sequence per sentence. Then we extract phrase-pairs together with the cooccuring English supertag sequence from this corpus via the same phrase extraction method used in the baseline model. This way we directly extend the baseline model described in section 3 with supertags both in the phrase translation table and in the language model. Next we define the probabilistic model that accompanies this syntactic enrichment of the baseline model.

Let ST represent a supertag sequence of the same length as a target sentence t. Equation (2) changes as follows:

$$\arg\max_{t} \sum_{ST} P(s \mid t, ST) P_{ST}(t, ST) \approx$$

$$\arg\max_{\langle t, ST\rangle} \overbrace{P(\phi_s \mid \phi_{t,ST})}^{TM\ w.sup.tags} \overbrace{P(O_s \mid O_t)^{\lambda_o}}^{distortion} \overbrace{P_{ST}(t, ST)}^{LM\ w.sup.tags} \overbrace{exp^{|t|\lambda_w}}^{word-penalty}$$

The approximations made in this formula are of two kinds: the standard split into components and the search for the most likely joint probability of a target hypothesis and a supertag sequence cooccuring with the source sentence (a kind of Viterbi approach to avoid the complex optimization involving the sum over supertag sequences). The distortion and word penalty models are the same as those used in the baseline PBSMT model.

Supertagged Language Model The 'language model' $P_{ST}(t, ST)$ is a supertagger assigning probabilities to sequences of word–supertag pairs. The language model is further smoothed by log-linear interpolation with the baseline language model over word sequences.

Supertags in Phrase Tables The supertagged phrase translation probability consists of a combination of supertagged components analogous to their counterparts in the baseline model (equation (3)), i.e. it consists of $P(s \mid t, ST)$, its reverse and a word-level probability. We smooth this probability by log-linear interpolation with the factored

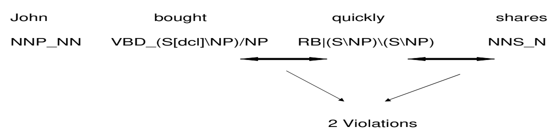

Figure 2: Example CCG operator violations: $V = 2$ and $L = 3$, and so the penalty factor is $1/3$.

backoff version $P(s \mid t)P(s \mid ST)$, where we import the baseline phrase table probability and exploit the probability of a source phrase given the target supertag sequence. A model in which we omit $P(s \mid ST)$ turns out to be slightly less optimal than this one.

As in most state-of-the-art PBSMT systems, we use GIZA++ to obtain word-level alignments in both language directions. The bidirectional word alignment is used to obtain lexical phrase translation pairs using heuristics presented in (Och & Ney, 2003) and (Koehn et al., 2003). Given the collected phrase pairs, we estimate the phrase translation probability distribution by relative frequency as follows:

$$\hat{P}_{ph}(s|t) = \frac{count(s,t)}{\sum_s count(s,t)}$$

For each extracted lexical phrase pair, we extract the corresponding supertagged phrase pairs from the supertagged target sequence in the training corpus (cf. section 5). For each lexical phrase pair, there is at least one corresponding supertagged phrase pair. The probability of the supertagged phrase pair is estimated by relative frequency as follows:

$$P_{st}(s|t,st) = \frac{count(s,t,st)}{\sum_s count(s,t,st)}$$

4.3 LMs with a Grammaticality Factor

The supertags usually encode dependency information that could be used to construct an 'almost parse' with the help of the CCG/LTAG composition operators. The *n*-gram language model over supertags applies a kind of statistical 'compositionality check' but due to smoothing effects this could mask crucial violations of the compositionality operators of the grammar formalism (CCG in this case). It is interesting to observe the effect of integrating into the language model a penalty imposed when formal compostion operators are violated. We combine the *n*-gram language model with a penalty factor that measures the number of encountered combinatory operator violations in a sequence of supertags (cf. Figure 2). For a supertag sequence of length (L) which has (V) operator violations (as measured by the CCG system), the language model P will be adjusted as $P* = P \times (1 - \frac{V}{L})$. This is of course no longer a simple smoothed maximum-likelihood estimate nor is it a true probability. Nevertheless, this mechanism provides a simple, efficient integration of a global compositionality (grammaticality) measure into the *n*-gram language model over supertags.

Decoder The decoder used in this work is Moses, a log-linear decoder similar to Pharaoh (Koehn, 2004), modified to accommodate supertag phrase probabilities and supertag language models.

5 Experiments

In this section we present a number of experiments that demonstrate the effect of lexical syntax on translation quality. We carried out experiments on the NIST open domain news translation task from Arabic into English. We performed a number of experiments to examine the effect of supertagging approaches (CCG or LTAG) with varying data sizes.

Data and Settings The experiments were conducted for Arabic to English translation and tested on the NIST 2005 evaluation set. The systems were trained on the LDC Arabic–English parallel corpus; we use the news part (130K sentences, about 5 million words) to train systems with what we call the *small* data set, and the news and a large part of the UN data (2 million sentences, about 50 million words) for experiments with *large* data sets.

The *n*-gram target language model was built using 250M words from the English GigaWord Corpus using the SRILM toolkit.[4] Taking 10% of the English GigaWord Corpus used for building our target language model, the supertag-based target language models were built from 25M words that were supertagged. For the LTAG supertags experiments, we used the LTAG English supertagger[5] (Bangalore

[4] http://www.speech.sri.com/projects/srilm/
[5] http://www.cis.upenn.edu/˜xtag/gramrelease.html

& Joshi, 1999) to tag the English part of the parallel data and the supertag language model data. For the CCG supertag experiments, we used the CCG supertagger of (Clark & Curran, 2004) and the Edinburgh CCG tools[6] to tag the English part of the parallel corpus as well as the CCG supertag language model data.

The NIST MT03 test set is used for development, particularly for optimizing the interpolation weights using Minimum Error Rate training (Och, 2003).

Baseline System The baseline system is a state-of-the-art PBSMT system as described in section 3. We built two baseline systems with two different-sized training sets: 'Base-SMALL' (5 million words) and 'Base-LARGE' (50 million words) as described above. Both systems use a trigram language model built using 250 million words from the English GigaWord Corpus. Table 1 presents the BLEU scores (Papineni et al., 2002) of both systems on the NIST 2005 MT Evaluation test set.

System	BLEU Score
Base-SMALL	0.4008
Base-LARGE	0.4418

Table 1: Baseline systems' BLEU scores

5.1 Baseline vs. Supertags on Small Data Sets

We compared the translation quality of the baseline systems with the LTAG and CCG supertags systems (LTAG-SMALL and CCG-SMALL). The results are

System	BLEU Score
Base-SMALL	0.4008
LTAG-SMALL	0.4205
CCG-SMALL	0.4174

Table 2: LTAG and CCG systems on small data

given in Table 2. All systems were trained on the same parallel data. The LTAG supertag-based system outperforms the baseline by 1.97 BLEU points absolute (or 4.9% relative), while the CCG supertag-based system scores 1.66 BLEU points over the baseline (4.1% relative). These significant improvements indicate that the rich information in supertags helps select better translation candidates.

POS Tags vs. Supertags A supertag is a complex tag that localizes the dependency and the syntax information from the context, whereas a normal POS tag just describes the general syntactic category of the word without further constraints. In this experiment we compared the effect of using supertags and POS tags on translation quality. As can be seen

System	BLEU Score
Base-SMALL	0.4008
POS-SMALL	0.4073
LTAG-SMALL	.0.4205

Table 3: Comparing the effect of supertags and POS tags

in Table 3, while the POS tags help (0.65 BLEU points, or 1.7% relative increase over the baseline), they clearly underperform compared to the supertag model (by 3.2%).

The Usefulness of a Supertagged LM In these experiments we study the effect of the two added feature (cost) functions: supertagged translation and language models. We compare the baseline system to the supertags system with the supertag phrase-table probability but without the supertag LM. Table 4 lists the baseline system (Base-SMALL), the LTAG system without supertagged language model (LTAG-TM-ONLY) and the LTAG-SMALL system with both supertagged translation and language models. The results presented in Table 4 indi-

System	BLEU Score
Base-SMALL	0.4008
LTAG-TM-ONLY	0.4146
LTAG-SMALL	.0.4205

Table 4: The effect of supertagged components

cate that the improvement is a shared contribution between the supertagged translation and language models: adding the LTAG TM improves BLEU score by 1.38 points (3.4% relative) over the baseline, with the LTAG LM improving BLEU score by

[6]http://groups.inf.ed.ac.uk/ccg/software.html

a further 0.59 points (a further 1.4% increase).

5.2 Scalability: Larger Training Corpora

Outperforming a PBSMT system on small amounts of training data is less impressive than doing so on really large sets. The issue here is scalability as well as whether the PBSMT system is able to bridge the performance gap with the supertagged system when reasonably large sizes of training data are used. To this end, we trained the systems on 2 million sentences of parallel data, deploying LTAG supertags and CCG supertags. Table 5 presents the comparison between these systems and the baseline trained on the same data. The LTAG system improves by 1.17 BLEU points (2.6% relative), but the CCG system gives an even larger increase: 1.91 BLEU points (4.3% relative). While this is slightly lower than the 4.9% relative improvement with the smaller data sets, the sustained increase is probably due to observing more data with different supertag contexts, which enables the model to select better target language phrases.

System	BLEU Score
Base-LARGE	0.4418
LTAG-LARGE	0.4535
CCG-LARGE	**0.4609**

Table 5: The effect of more training data

Adding a grammaticality factor As described in section 4.3, we integrate an impoverished grammaticality factor based on two standard CCG combination operations, namely Forward and Backward Application. Table 6 compares the results of the baseline, the CCG with an *n*-gram LM-only system (CCG-LARGE) and CCG-LARGE with this 'grammaticalized' LM system (CCG-LARGE-GRAM). We see that bringing the grammaticality tests to bear onto the supertagged system gives a further improvement of 0.79 BLEU points, a 1.7% relative increase, culminating in an overall increase of 2.7 BLEU points, or a 6.1% relative improvement over the baseline system.

5.3 Discussion

A natural question to ask is whether LTAG and CCG supertags are playing similar (overlapping, or conflicting) roles in practice. Using an oracle to choose the best output of the two systems gives a BLEU score of 0.441, indicating that the combination provides significant room for improvement (cf. Table 2). However, our efforts to build a system that benefits from the combination using a simple log-linear combination of the two models did not give any significant performance change relative to the baseline CCG system. Obviously, more informed ways of combining the two could result in better performance than a simple log-linear interpolation of the components.

System	BLEU Score
Base-LARGE	0.4418
CCG-LARGE	0.4609
CCG-LARGE-GRAM	**0.4688**

Table 6: Comparing the effect of CCG-GRAM

Figure 3 shows some example system output. While the baseline system omits the verb giving "the authorities that it had...", both the LTAG and CCG found a formulation "authorities reported that" with a closer meaning to the reference translation "The authorities said that". Omitting verbs turns out to be a problem for the baseline system when translating the notorious verbless Arabic sentences (see Figure 4). The supertagged systems have a more grammatically strict language model than a standard word-level Markov model, thereby exhibiting a preference (in the CCG system especially) for the insertion of a verb with a similar meaning to that contained in the reference sentence.

6 Conclusions

SMT practitioners have on the whole found it difficult to integrate syntax into their systems. In this work, we have presented a novel model of PBSMT which integrates supertags into the target language model and the target side of the translation model.

Using LTAG supertags gives the best improvement over a state-of-the-art PBSMT system for a smaller data set, while CCG supertags work best on a large 2 million-sentence pair training set. Adding grammaticality factors based on algebraic compositional operators gives the best result, namely 0.4688 BLEU, or a 6.1% relative increase over the baseline.

Reference: *The authorities said he was allowed to contact family members by phone from the armored vehicle he was in.*
Baseline: *the authorities that it had allowed him to communicate by phone with his family of the armored car where*
LTAG: *authorities reported that it had allowed him to contact by telephone with his family of armored car where*
CCG: *authorities reported that it had enabled him to communicate by phone his family members of the armored car where*

Figure 3: Sample output from different systems

Source*: wmn AlmErwf An Al$Eb AlSyny mHb llslAm .*
Ref*: It is well known that the Chinese people are peace loving .*
Baseline: *It is known that the Chinese people a peace-loving .*
LTAG*: It is known that the Chinese people a peace loving .*
CCG*: It is known that the Chinese people are peace loving .*

Figure 4: Verbless Arabic sentence and sample output from different systems

This result compares favourably with the best systems on the NIST 2005 Arabic–English task. We expect more work on system integration to improve results still further, and anticipate that similar increases are to be seen for other language pairs.

Acknowledgements

We would like to thank Srinivas Bangalore and the anonymous reviewers for useful comments on earlier versions of this paper. This work is partially funded by Science Foundation Ireland Principal Investigator Award 05/IN/1732, and Netherlands Organization for Scientific Research (NWO) VIDI Award.

References

S. Bangalore and A. Joshi, "Supertagging: An Approach to Almost Parsing", *Computational Linguistics* **25**(2):237–265, 1999.

J. Chen, S. Bangalore, and K. Vijay-Shanker, "Automated extraction of tree-adjoining grammars from treebanks". *Natural Language Engineering*, 12(3):251–299, 2006.

D. Chiang, "A Hierarchical Phrase-Based Model for Statistical Machine Translation", in *Proceedings of ACL 2005*, Ann Arbor, MI., pp.263–270, 2005.

S. Clark and J. Curran, "The Importance of Supertagging for Wide-Coverage CCG Parsing", in *Proceedings of COLING-04*, Geneva, Switzerland, pp.282–288, 2004.

J. Hockenmaier, *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*, PhD thesis, University of Edinburgh, UK, 2003.

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.

P. Koehn, "Pharaoh: A Beam Search Decoder for phrase-based Statistical Machine Translation Models", in *Proceedings of AMTA-04*, Berlin/Heidelberg, Germany: Springer Verlag, pp.115–124, 2004.

P. Koehn, F. Och, and D. Marcu, "Statistical Phrase-Based Translation", in *Proceedings of HLT-NAACL 2003*, Edmonton, Canada, pp.127–133, 2003.

D. Marcu, W. Wang, A. Echihabi and K. Knight, "SPMT: Statistical Machine Translation with Syntactified Target Language Phrases", in *Proceedings of EMNLP*, Sydney, Australia, pp.44–52, 2006.

D. Marcu and W. Wong, "A Phrase-Based, Joint Probability Model for Statistical Machine Translation", in *Proceedings of EMNLP*, Philadelphia, PA., pp.133–139, 2002.

F. Och, "Minimum Error Rate Training in Statistical Machine Translation", in *Proceedings of ACL 2003*, Sapporo, Japan, pp.160–167, 2003.

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

K. Papineni, S. Roukos, T. Ward and W-J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation", in *Proceedings of ACL 2002*, Philadelphia, PA., pp.311–318, 2002.

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 Kaufmann, pp.267–296, 1990.

M. Steedman, *The Syntactic Process*. Cambridge, MA: The MIT Press, 2000.

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.